The Dissertation Committee for Joshua David Harguess
certifies that this is the approved version of the following dissertation:

Face Recognition from Video

Committee:

J. K. Aggarwal, Supervisor

Al Bovik

Joydeep Ghosh

Kristen Grauman

Michael Ryoo

Face Recognition from Video

by

Joshua David Harguess, B.S.E., M.S.C.A.M.

DISSERTATION

Presented to the Faculty of the Graduate School of

The University of Texas at Austin

in Partial Fulfillment

of the Requirements

for the Degree of

DOCTOR OF PHILOSOPHY

THE UNIVERSITY OF TEXAS AT AUSTIN

December 2011

To my wife Désirée,

to my parents,

to my brother,

and to my grandparents

Acknowledgments

This thesis is based on work I have done as a Ph.D. student in the Electrical and Computer Engineering (ECE) Department and as a member of the Computer and Vision Research Center (CVRC) at the University of Texas at Austin under the direction of Prof. J. K. Aggarwal. It has been supported in part by the ECE department, the General Engineering (GE) teaching assistant program, and the Science, Mathematics and Research for Transformation (SMART) Scholarship for Service Program. I am truly grateful for each of these sources of financial support and the unique opportunities that have come with each of them.

I want to first thank my advisor, Prof. Aggarwal, for giving me the chance to do this work, for the encouragement, and for the guidance. Without his tireless support and advice, this dissertation would not have been possible. His continued dedication to a field that he helped pioneer almost 50 years ago is an inspiration.

I want to thank Prof. Bovik for his thoughtful discussions and enthusiasm for the fields of computer vision and image processing, particularly in my early years as a Ph.D. student. Thanks to Prof. Grauman, Prof. Ghosh and Dr. Ryoo for their helpful comments and suggestions during the course of my Ph.D. and for taking the time to participate as committee members.

Thanks to my fellow graduate students in the CVRC lab and their collaboration; Birgi Tamersoy, Chia-Chih Chen, Jong Taek Lee, Suyog Jain, Dr. Elden Yu, and Dr. Michael Ryoo. Special thanks to Dr. Changbo Hu for his helpful collaboration over the past couple of years. My interactions with him were integral to the work in this dissertation.

I would like to also thank Melanie Gulick and Trudie Redding for their help in navigating the complex systems of the ECE department, the Graduate School, and funding resources. I am truly grateful for their help.

I was fortunate enough to work with Livy Knox in the General Engineering (GE) program for two years as a teaching assistant and I am forever grateful to her and the GE program for developing my skills as an instructor and for fostering my love for teaching.

My incredibly supportive friends in Austin, Phoenix, San Diego and elsewhere, are too numerous to mention here, but they are truly a source of inspiration and I am thankful for every one of them.

I have been blessed with an extremely loving and supportive family. Many thanks to my mother and father who are always there to support me in all my endeavors. A special thanks to my mother for agreeing to read and edit this dissertation. Thanks to my brother, my sister-in-law and my two beautiful nieces. Thanks to my wife for her love and support throughout my graduate school years. I could not have done this without her. Thanks to my two dogs that show me unconditional love as only dogs can.

Finally, I would like to thank Dr. Alvin Swimmer who piqued my interest in the mathematical sciences at Arizona State University during my undergraduate years as an Electrical Engineering student. Dr. Swimmer started every class with a quotation, so I will end my acknowledgements with a favorite of my own.

> New knowledge is the most valuable commodity on earth. The more truth we have to work with, the richer we become.
> -Kurt Vonnegut, Jr., *Breakfast of Champions*

Face Recognition from Video

Publication No. _____

Joshua David Harguess, Ph.D.
The University of Texas at Austin, 2011

Supervisor: J. K. Aggarwal

While the area of face recognition has been extensively studied in recent years, it remains a largely open problem, despite what movie and television studios would leave you to believe. Frontal, still face recognition research has seen a lot of success in recent years from many different researchers. However, the accuracy of such systems can be greatly diminished in cases such as increasing the variability of the database, occluding the face, and varying the illumination of the face. Further varying the pose of the face (yaw, pitch, and roll) and the face expression (smile, frown, etc.) adds even more complexity to the face recognition task, such as in the case of face recognition from video. In a more realistic video surveillance setting, a face recognition system should be robust to scale, pose, resolution, and occlusion as well as successfully track the face between frames. Also, a more advanced face recognition system should be able to improve the face recognition result by utilizing the information present in multiple video cameras.

We approach the problem of face recognition from video in the following manner. We assume that the training data for the system consists of only still image data, such as passport photos or mugshots in a real-world system. We then transform the problem of face recognition from video to a still face recognition problem. Our research focuses on solutions to detecting, tracking and extracting face information from video frames so that they may be utilized effectively in a still face recognition system.

We have developed four novel methods that assist in face recognition from video and multiple cameras. The first uses a patch-based method to handle the face recognition task when only patches, or parts, of the face are seen in a video, such as when occlusion of the face happens often. The second uses multiple cameras to fuse the recognition results of multiple cameras to improve the recognition accuracy. In the third solution, we utilize multiple overlapping video cameras to improve the face tracking result which thus improves the face recognition accuracy of the system. We additionally implement a methodology to detect and handle occlusion so that unwanted information is not used in the tracking algorithm. Finally, we introduce the average-half-face, which is shown to improve the results of still face recognition by utilizing the symmetry of the face. In one attempt to understand the use of the average-half-face in face recognition, an analysis of the effect of face symmetry on face recognition results is shown.

Table of Contents

List of Tables

List of Figures

Chapter 1

Introduction

If you are a fan of science fiction, chances are you have seen the television show "Knight Rider" or movies such as the James Bond movie "Quantum of Solace" or "Mission Impossible". In these popular shows and others, face recognition appears as a completely solved technology that can be used with extremely high accuracy to identify "the bad guy". However, the notion that face recognition is solved, especially from realistic video, is greatly exaggerated.

Face recognition is an important area of computer vision research and has gained significant interest in recent years. Efforts in improving security, such as automatic surveillance and the use of biometrics in identification, are partly responsible for this increased interest. However, several challenges remain in improving the accuracy of face recognition under illumination changes, variations in pose, occlusions (including self-occlusion), and image resolution. Many face recognition algorithms have been developed and each has its strengths and weaknesses.

Research into face recognition is decidedly multidisciplinary in its approach and its applications. Researchers from the fields of Psychology, Neu-

roscience, Computer Science, Image and Video Processing, Machine Learning, Data Mining, Applied Mathematics, and others have all made significant contributions to our understanding of face recognition as it relates to human ability as well as automatic face recognition from machines. In the field of computer vision, we are mostly concerned with improving the ability of machines to automatically recognize human faces using visual data.

1.1 Face Recognition from Still Faces and Video

There are two main types of visual data sources used in face recognition. The most familiar is that of still (not moving), mostly frontal faces. Many databases containing still faces have been built and released to the face recognition community for use in research. Countless algorithms have been implemented and evaluated on this data and significant increases in accuracy have been achieved. We build upon and utilize these advances in the research discussed in this dissertation. More recently, the visual data source of video has been of interest to face recognition researchers. Video face data can be thought of as a sequence of still face images. Therefore, still face recognition methods can be applied to video face data. The interest in video face data is mostly due to the wide availability of inexpensive and high quality video cameras, such as webcams, hand-held video cameras, and surveillance video cameras. These two types of visual data for face recognition will be discussed in greater detail below.

1.1.1 Still Face Recognition

As previously mentioned, still face recognition is the most common type of face recognition research. The data usually consists of a mostly frontal face image, such as that found in Figure 1.1, which is from the popular Yale Face database [1].

Face recognition of still faces generally involves a process such as that depicted in Figure 1.2. Given a particular database used for the face recognition task, the images are first preprocessed. Then, features are extracted from the image and stored in a 'feature vector', which describe the content of the face in the image used for recognition. Then the test images in the database are classified based on their feature vectors to the most likely candidate in the training set. Finally, the overall performance of the face recognition algorithm is evaluated using several standard methods so that comparisons can be made between different face recognition methods. Each of these blocks will be explained in further detail in Chapter 2.

1.1.2 Face Recognition from Video

Recognizing faces from video may be much more challenging than recognizing faces from still face images. Several challenges are imposed when trying to recognize faces from video, such as image resolution, motion blur, varying illumination, and nonfrontal poses of the face. Three images from an example surveillance video containing a face with the previously mentioned challenges appear in Figure 1.3. In the left column of the figure, the example

Figure 1.1: Example Still Face Image from the Yale Face database

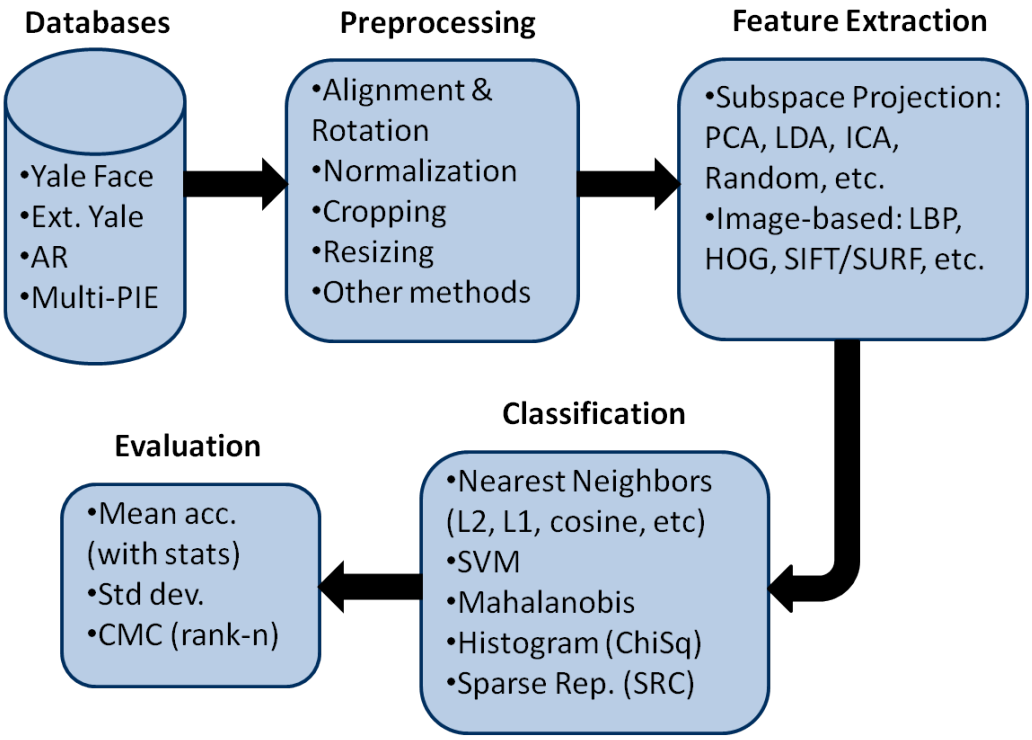


Figure 1.2: Overview of Face Recognition From Still Face Data

frames are shown. In the right column of the figure, the face detection results of each of the frames is shown. On off-the-shelf face detector (such as that found in OpenCV [11]) has difficulty even detecting the face in the third frame because of the face pose and the motion blur.

Because there are a greater number of challenges to confront in a face recognition from video problem, the approach to the solution is more complex. Figure 1.4 gives an overview of the way we approach the face recognition from video problem. We consider the problem of face recognition from still face images as a subproblem of face recognition from video. By doing so, we can utilize the extensive research that has been done in face recognition from still face images in our research. Therefore, the bottom row of Figure 1.4 is essentially the same as that of Figure 1.2. The major difference is that we must process the video data to detect the face in the image sequence, track the face throughout the image sequence and finally extract meaningful face texture that can be used in the still face recognition process.

1.2 Problem Statement

We consider the problem of face recognition from video in the following context. First, we assume that the training data comes from very few examples of still face data for each of the individuals that we wish to recognize. Practical examples of such data would be driver's license photos, passport photos, mugshots, etc. We limit ourselves to this type of training data because it is the most common type of face data that is readily available to security personnel.

Figure 1.3: Three sample frames from example surveillance video (left column) and the face detection result of each frame (right column)

That is not to say that our framework could not be used on video face training data, but additional work would be needed to extract the training face data from the video.

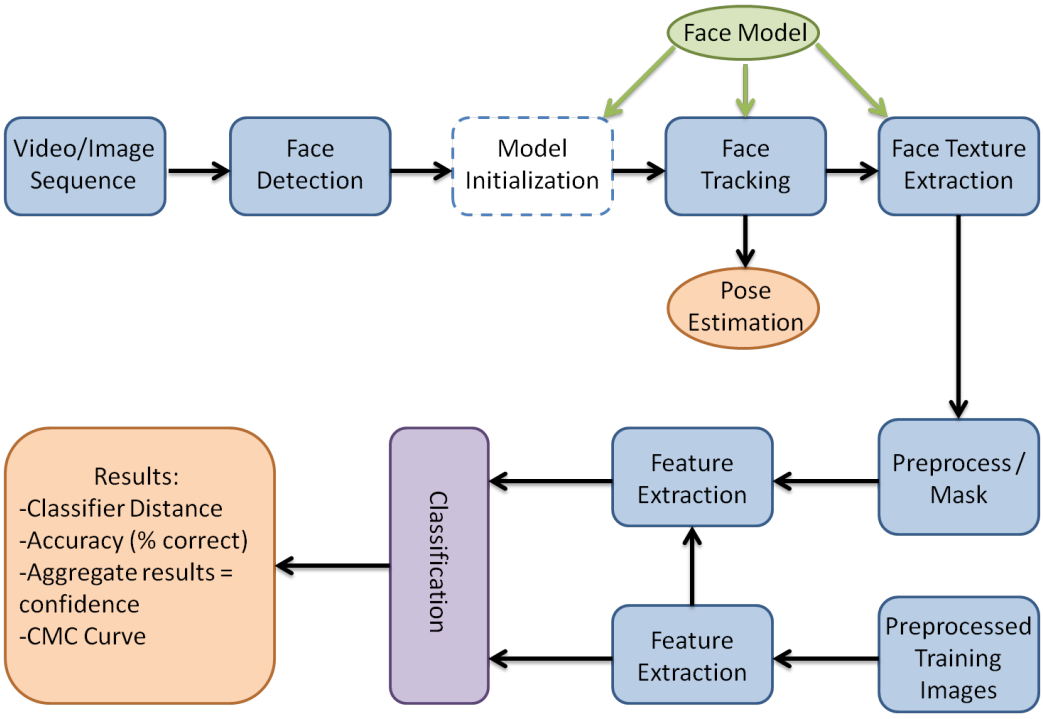


Figure 1.4: Overview of Face Recognition From a Single Video

Second, since the training data of the overall face recognition algorithm is that of still face data, we must transform the face video data to still face data. The heart of our research lies in this task and it involves accurate face detection, face tracking and face texture extraction. If we are able to extract meaningful texture from each frame of the video, then we are able to better recognize the face of the individual present in that video.

Finally, we approach the problem from a real-world surveillance context by assuming that multiple video cameras may be present and observing the scene simultaneously. Multiple cameras may have overlapping views of the face and therefore will have multiple views of the face that may assist in identifying the individual. Also, these overlapping cameras may provide additional benefits, such as improving the tracking of the face.

Many formulations of face recognition research exist and we have chosen above focus for the research in this dissertation.

1.3 Challenges

The challenges in the problem of face recognition from video are many. As previously mentioned, illumination, image resolution, motion blur, non-frontal face pose and other issues all contribute to the problem in several ways. First, it may be difficult or impossible to detect the face in a given frame. Second, given a successful face detection in the current frame, it may be difficult to track the face in successive frames leading to incorrect detections and therefore unreliable face texture. Even if the face is detected successfully

and meaningful face texture is extracted from the image, that does not directly translate to the successful recognition of the individual in a given frame. We provide our solutions to the above challenges in this dissertation.

1.4 Our Contributions

We present four main approaches to assist in the face recognition from video problem. This will also serve as the outline to the dissertation.

1. In Chapter 3, we consider the problem of recognizing an individual in a video in which only face patches of the individual exist, such as when the face is partially occluded. In these types of video, the full frontal face of the individual may not be present in any of the frames. We provide a solution to such a problem by coarsely aligning the face patches to a template face and then stitching the patches to reconstruct the face from several frames in the video. We then use the reconstructed face in the still face recognition framework [42].

2. Next, we consider the problem of face recognition from multiple overlapping video cameras. Our solution, presented in Chapter 4, involves independently tracking the face from each of the cameras and extracting face texture from each of the cameras simultaneously [37]. We then perform still face recognition on the extracted face textures and fuse the recognition results to obtain a significant improvement to overall face recognition accuracy. We also introduce a confidence measure of the ac-

curacy by aggregating the results of the recognition within a particular video sequence.

3. We build upon the concept of utilizing multiple cameras for face recognition in Chapter 5. In addition to fusing the recognition results from multiple cameras, we also present a novel face tracking algorithm that utilizes the multiple views of the face to calculate a joint estimation of the face motion between frames. We also address the problem of camera occlusion and provide an occlusion detection algorithm to remove unwanted information in the joint motion estimation process [39]. These advances lead to a significant improvement in face tracking and provide a robust solution to the problem. We demonstrate the solution on the problem of pose estimation of the face as well as face recognition from multiple video cameras [38].

4. Finally, in Chapter 6, we introduce the average-half-face, which takes advantage of the symmetry of the face in the still face recognition process. We present the results of using the average-half-face on several face recognition databases and show a significant improvement in face recognition accuracy over using the original full face. In an effort to discover why the average-half-face improves face recognition accuracy, we also present an analysis of the effect of face symmetry on face recognition.

We summarize and conclude these methods in Chapter 7.

Chapter 2

Related Work

The field of face recognition research is vast and varied. For the purposes of this dissertation, we will focus our attention on work related to face recognition from still images and from video. In the case of still images, we are mostly concerned with two-dimensional (2D) images, since that is the type of image we would expect in a surveillance setting. However, we will demonstrate some results that utilize three-dimensional face data (or range data) in Chapter 6.

2.1 Still Face Recognition

Still face recognition from 2D images is by far the most well-known and most extensively researched field of face recognition research. As Chellappa *et al.* noted recently [15], machine face recognition algorithms can outperform humans, even across changes in illumination, on frontal still face images. However, humans still far outperform machines when the face portrays changes in pose, illumination, blur, resolution or any combination of the above. Faces that appear in surveillance video, for example, often exhibit the types of issues outlined above. Therefore, approaches used to tackle these problems in

still face recognition may be used in face recognition from video as well.

An overview of a typical still face recognition algorithm is show in Figure 1.2. In this dissertation, we will attempt to utilize the research that has been done in the area for our own work. In other words, our approach to face recognition from video is to transform the problem to a still face recognition problem. Therefore, we are particularly interested in utilizing the 'Feature Extraction' and 'Classification' blocks of Figure 1.2 for our purposes.

There are two main types of still face recognition that we will describe here; holistic (global) and feature-based (local). Each approach differs mostly in the feature representation of the face that is used for recognition. However, the classification stage of the algorithm is largely the same and will be covered separately in Section 2.1.3.

2.1.1 Holistic Still Face Recognition

Holistic (or global) face recognition is based on using the entire region of the face for recognition. Feature extraction methods in this type of face recognition are typically subspace projection techniques, meaning that the high-dimensional data in the original image (number of pixels wide x number of pixels tall) is projected in to a low-dimensional 'face space'. The two most popular methods are known as 'eigenfaces' [79] and 'fisherfaces' [8] and will be introduced below along with several other popular holistic face recognition methods. A thorough survey of these methods is presented in [93].

2.1.1.1 Eigenfaces

Eigenfaces [79], one of the first successful face recognition algorithms, is based on principal components analysis (PCA), a well-known tool for exploratory data analysis and dimension reduction technique. Eigenfaces is a subspace projection face recognition method that relies on computing the PCA of a training set which will return a set of orthogonal basis vectors that maximize the variance in the training data and in turn form the 'face space'. Each image training and testing image is then projected into the 'face space' and the test vectors are classified as the most likely training vector. PCA is an unsupervised technique, so the method does not rely on class information.

2.1.1.2 Fisherfaces

Fisherfaces is the direct use of (Fisher) linear discriminant analysis (LDA) to face recognition [8]. In eigenfaces, the variance between vectors is used to find a linear subspace for projection, without taking into consideration the class associations of each training vector. In LDA, the class information is explicitly used to form a linear subspace. The purpose of LDA is to maximize the objective function:

$$J(w) = \frac{w^T S_B w}{w^T S_W w} \tag{2.1}$$

where S_B and S_W are the *between class scatter* and the *within class scatter* matrices respectively and where w is the normal vector to the discriminant hyperplane. The solution can be posed as the eigenvalue problem

$$S_B^{\frac{1}{2}} S_W^{-1} S_B^{\frac{1}{2}} v = \lambda v \tag{2.2}$$

by defining $v = S_B^{\frac{1}{2}} w$. We desire the eigenvectors that correspond to the largest eigenvalues for our solution. Projecting our training and test vectors into this new subspace, we can then classify the test images.

2.1.1.3 Multilinear Principal Components Analysis

One extension of PCA is that of applying PCA to tensors or multilinear arrays which results in a method known as multilinear principal components analysis (MPCA) [59]. Since a face image is most naturally a multilinear array, meaning that there are two dimensions describing the location of each pixel in a face image, the idea is to determine a mulitlinear projection for the image, instead of forming a one-dimensional (1D) vector from the face image and finding a linear projection for the vector. It is thought that the multilinear projection will better capture the correlation between neighborhood pixels that is otherwise lost in forming a 1D vector from the image.

A further extension of MPCA is to use linear discriminant analysis on the projected multilinear arrays to perform feature selection, which results in MPCA+LDA.

2.1.1.4 Independent Components Analysis

When applying PCA to a set of face images, we are finding a set of basis vectors using lower order statistics of the relationships between the pix-

els. Specifically, we maximize the variance between pixels to separate linear dependencies between pixels. Independent components analysis (ICA) is a generalization of PCA in that it tries to identify high-order statistical relationships between pixels to form a better set of basis vectors. As described in [5], the pixels are treated as random variables and the face images as outcomes. In a similar fashion to PCA and LDA, once the new basis vectors are found, the training and test data are projected into the subspace and used for classification.

2.1.2 Feature-Based Still Face Recognition

Feature-based (or local) face recognition methods use various descriptions of the face for recognition. For instance, purely geometric approaches use the distances and ratios of distances between landmarks on the face (such as corners of the eyes and mouth). Distances between points can be measured in terms of Euclidean (2D or 3D), or more recently, geodesic (3D) distances [32]. Three recent advances in feature-based still face recognition are presented below.

2.1.2.1 Elastic Bunch Graph Matching

The elastic bunch graph matching (EBGM) algorithm, proposed by Wiskott *et al.* [82], is one of the most successful feature-based face recognition algorithms. Gabor wavelets of different scales and orientations are used to provide a local description of landmarks on the face. Then, a graph is built

from connecting the nodes on the landmarks of the face to form a face bunch graph (FBG). Classification for EGBM is done by computing a similarity score between the FBG of a test image and the FBG of the training images.

2.1.2.2 Local Binary Patterns

Several researchers [3, 28, 52, 89] have utilized Local Binary Patterns (LBP) to capture face features useful in face recognition. LBP operates on gray-scale texture to characterize the local spatial structure of the image texture. A 2-bit pattern code is computed by comparing a central pixel with its neighbours:

$$LBP = \sum s(p_n - p_c)2^N \tag{2.3}$$

where p_c is the gray value of the central pixel, p_n is the value of its neighbors, s is evaluated to be 0 if $p_c > p_n$ and 1 otherwise and N is the total number of neighbors involve in the computation. Another parameter, R, controls the radius of the neighborhood. After computing the LBP pattern of each pixel in the image, a histogram is built to represent the whole texture image. Then, a comparison of sample and model histograms is a done, typically using a nonparametric statistical test of goodness-of-fit.

2.1.2.3 Face Attributes

Recent advances have taken advantage of extremely large training databases that have been manually labeled with attributes, such as the work of Kumar

et al. [47]. In their work, overall face similarity scores are computed based on scores computed on patches of the face. Each patch is compared against a massive number of labeled exemplar patches and attributes are assigned to the patch based on these comparisons. This methodology has shown to achieve high accuracy on face databases such as Labeled Faces in the Wild [43]. However, extending the method to work on face recognition from video with few training images is not trivial, or even possible at this time.

2.1.3 Classification

Once the training and testing data have been projected into 'face space', a classification algorithm is used to assign labels to the unknown testing data. The nearest neighbors algorithm (NN) [24] is frequently used and is a powerful, but simple method. The distance is measured between a test sample and the training samples and the label of the training sample with the minimum distance is used as the label for the test sample. Many types of distances can be used, such as Euclidean distance (L2), Manhattan distance (L1), and the cosine distance. Additionally, histogram-based distances are used to classify features which are inherently statistical in nature, such as the chi-square distance or the Bhattacharyya divergence [9]. Two more recent advances in classification are presented below.

2.1.3.1 Support Vector Machine

Support Vector Machines (SVM) are a type of binary classifier that are designed to maximize the *margin* of the decision boundary between positive and negative examples, or support vectors [70]. This amounts to finding the most informative positive and negative support vectors and maximizing the margin between them to form the optimal decision boundary for classifying new vectors.

2.1.3.2 Sparse Representation

Many recent methods utilize the sparse representation classification (SRC) methodology for face recognition introduced by Wright *et al.* [85]. With the assumption that training and testing images are well-aligned, a test image of a subject in the training data can be represented by a sparse combination of the training images.

An assumption is made that the face images belonging to the same person all lie on a low-dimensional linear subspace, represented by matrices $A_1, A_2, ..., A_k$ for each subject, and that each column in A_j is a vector formed from a training image of subject j. Now a test sample u from subject j can be expressed as a (sparse) linear combination of the columns of A_j. We let

$$A = [A_1 A_2 ... A_k]. \tag{2.4}$$

Ideally, we would like to solve the following optimization problem to

get the sparsest solution to $Ax = u$:

$$x_0 = argmin||x||_0 \quad subject \quad to \quad Ax = u. \tag{2.5}$$

However, the solution with respect to the l^0 norm is NP-hard. Therefore, an alternative to the above optimization is to replace the l^0 norm with the l^1 norm. This new optimization problem can be solved using linear programming methods in polynomial time and it is equal to the original solution if x_0 is 'sparse enough'. Once the optimization problem is solved for a given test subject, we classify the test vector u to the training subject with the most (and largest) coefficients.

2.2 Face Recognition from Video

Face recognition from video has received much interest in recent times. This is likely due to heightened security and the availability of inexpensive surveillance cameras. Also, face recognition from video may produce better overall accuracy since a video will have many frames of a subject's face instead of just a few examples.

There are two main problems within the area of face recognition from video, as described by Wang *et al.* in [80]. The first, and most common problem is matching a face that appears in video to still face images (still-video), such as passport or driver's license photographs. The second is matching a face that appears in video to a face previously seen videos (video-video). To further

clarify, still-video face recognition from video methods can utilize still images or video frames for training, but they do not rely training video to be present, such as the case in video-video face recognition from video methods. Our work is focused mainly in the area of the first problem (still-video) since we are assuming a small training size of still face images. However, our methods could be adapted to work in the second problem (video-video) as well. We will review previous work in both problem areas.

2.2.1 Still-Video Face Recognition

The research presented in this dissertation is most related to the still-video face recognition problem. In this problem, only a few still images are given as gallery and/or training images to build the face recognition system. Then video sequences, such as that from a surveillance camera, are used in the testing phase. An overview of such a system is given in Figure 1.4.

One approach that several researchers have used [27, 40, 44, 49, 66, 77] is to apply Active Appearance Models (AAM) to track the face in input video and warp the face to a mean shape used for recognition. The main drawback to these approaches is in the use of AAM for face tracking. First of all, training most AAM-based face models is very time consuming and does not usually produce acceptable results on faces that were not in the training set. Second, even when generic AAM models are able to track new faces, the tracking result itself is only useful in very small pose changes of the face which limits its use for face recognition from real world video such as that of a surveillance camera.

A popular approach to face recognition from video is to apply a methodology for selecting 'good' frames or a subsets of the video that may be used for recognition in place of the original video sequence. In general, these methods are complimentary to our research and may be used to further improve the face recognition result. Wong *et al.* [84] introduce a patch-based probabilistic image quality assessment for method for selecting the face images that may be more suitable for recognition. Their method is based on comparing local patches of a face of the input video sequence to an 'ideal face'. They also introduce the ChokePoint Dataset [83], which is a multi-camera video database of 29 different subjects entering and leaving several doorways in realistic surveillance settings. Even though it has just been recently released to the public, this is the first freely available database of its kind to our knowledge and may be helpful in testing still-video and video-video face recognition from video algorithms.

In a similar approach, Xie *et al.* [87, 88] introduce a reliability-based method to select the most appropriate images for face recognition. In their work, they utilize multiple cameras and select the best camera for face recognition based on a template image. Scores from the cameras are combined using majority voting and classifier distances. In our work, we additionally fuse the estimated pose of the face to improve the face recognition result.

To our knowledge, the only other multi-camera face recognition approach, and possibly the first, is from the work of Stillman *et al.* [75]. They introduce a method that utilizes two types of cameras; one for person tracking

that has a broad view of the scene and a pan-tilt-zoom (PTZ) camera that provides face data for recognition. The main drawback of this method is the assumption that nearly frontal faces will be present in the PTZ camera.

Brute force methods that use the video sequence essentially as a set of training still images also exist, such as in [7]. Our methodology is most similar to this approach except for two main differences. First, we assume we have a small sample size of training images available. Second, our methodology takes advantage of pose of the face by modeling it directly and using it in the recognition stage.

Chellappa *et al.* [16] introduce two Bayesian methods for face recognition from video based on a time series state space model. The two methods are both used to implement the Sequential Importance Sampling (SIS) technique which generates a numerical solution to estimating the posterior distribution of the test subject's identity.

2.2.2 Video-Video Face Recognition

Several researchers have proposed solutions to the video-video face recognition problem. These methods are not directly comparable to our work since they inherently rely on video data for training.

Li *et al.* [51] first constructs a facial identity surface for a given subject using a multi-view face model to capture the spatio-temporal information of the training video. The detected faces in the input video are then warped to the mean shape with the frontal view using the multi-view model. Kernel

discriminant analysis (KDA) is then used for recognition of the identity surface.

In the work of Lee *et al.* [50], training video of each of the subjects is used to build a probabilistic appearance manifold. Face recognition from the test videos is then performed using a maximum a posteriori formulation by integrating the likelihood that the input image comes from a particular pose manifold along with the transition probability between frames.

Liu and Cheng [55] use a Hidden Markov Model (HMM) to learn the statistics of the training video sequences for each subject along with the temporal dynamics. The HMM is then adapted to the test video sequences and likelihood scores are used for classification.

Zhou *et al.* [95] use a probabilistic model to recognize faces from video using both still images and video as the gallery.

See and Eswaran [71] propose using a spatio-temporal hierarchical agglomerative clustering (STHAC) methodology to automatically extract face exemplars from video sequences for recognition. Classification is performed using a Bayes framework with probabilistic voting to combine the results across different frames of the video sequence.

There are also methodologies that seek to solve the problem of person identification through multibiometrics, which is when more than one modality of data is used for recognition. For instance, Shakhnarovich *et al.* [73] integrate face and gait data from videos to perform person identification from video. However, their methodology requires full-body tracking and a sufficient

number of occlusion-free frames to build their face and gait model.

2.3 Summary

While many methods exist to tackle the problem of face recognition from video or related problems, it is still a largely open area of research. We present novel solutions to face recognition from video that largely compliment the methods outlined above.

Chapter 3

Patch-based Face Recognition from Video

In this chapter, we will introduce our work on patch-based video face recognition. Figure 3.1 displays an overview of our system. First, face patches from the image sequence are aligned and stitched together to form a reconstructed face. Then, the reconstructed face is used in a frontal face recognition algorithm. The sections below will explain these concepts in further detail.

3.1 Introduction

In general, video provides more information for recognition as compared to a still image. However, several challenging problems still remain unsolved, such as changes in illumination, pose, and occlusion. One critical problem is matching corresponding pixels from overlapping face regions from successive images in a video sequence under changes in illumination, pose, and occlusion. This is a serious problem when only part of the face region is shown and the same region may appear in different poses and scales. One desires a method to correspond the parts of the full faces or face patches, collect the face patches from video, and construct a full face or as much of a face region as possible. The recognition is then based on the available face region collected. In this paper,

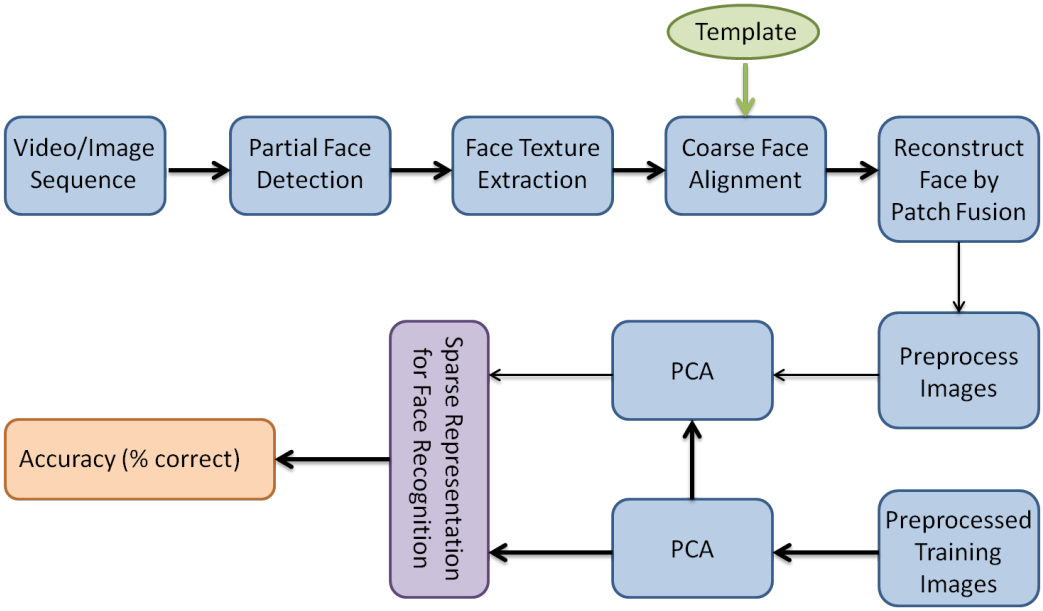


Figure 3.1: Overview of Patch-Based Face Recognition From Video

we propose a patch-based method to recognize a face from video. Our patch-based method provides a correspondence framework to organize available face regions to the correct location in the full face template image by using image registration [60]. An image stitching method is used to construct the full face image at the pixel level. We employ face recognition via sparse representation [85] to recognize the reconstructed faces. The region that we are unable to recover from the video sequence is treated as an occluded region.

Comparing to existing patch-based methods, in [4] patch correspondence is based on many patches from one face to another face in different views. Our method is based on a single patch correspondence to a full face template image.

When comparing our method to face reconstruction methods, [92] needs a 3D setting and is able to reconstruct very high quality 3D faces. In [48], face reconstruction is based on a cylindrical face model. Our method is based on the reconstruction from 2D patches and is more flexible and suitable for a video-based face recognition task. Among 2D based face methods, our method differs in the alignment of patches and the refinement of the patch alignment with an image stitching technique which decomposes the correspondence process into two steps and is likely to be more robust and efficient.

Also, any still face recognition method can be employed after reconstructing the full face image. This enables the use of many existing still face recognition algorithms for the patch-based face recognition from video system. We employ face recognition via sparse representation [85] to handle the missing data encountered in the proposed framework. Since capturing a single full face image from video is not guaranteed, we only reconstruct as much of the face as possible from the video sequence. Normally, the reconstructed results will cover most of the face, but some regions of the face may be left blank. The sparse representation method provides a powerful tool to handle the regions of the face that cannot be recovered.

Several experiments are conducted to test the proposed method. The first experiment is on a still image database. We partition the full face region into random subregions, or face patches, and use them to reconstruct the full face. We estimate the reconstruction error and the recognition accuracy of the reconstructed faces. The experiments show that we can successfully

reconstruct the face for recognition.

In another experiment using video that is generated in our lab, a face in the video may appear in various poses. We transform the patch to its correct location on the template face image and stitch the regions into a full face image. It is shown that a full face, or most of a full face, may be reconstructed with high accuracy. Sparse representation is used to classify the reconstructed faces.

3.2 Face Reconstruction From Video

We model a partial face image as a patch that is taken from a full face image. This task has two steps. First we align the face patch to the frontal template face, which is simply an example 2D frontal face from the training images. Next, we stitch several partial face patches together to reconstruct a seamless full face.

3.2.1 Face Patch Alignment

The first step of our face patch alignment algorithm is to locate face patches in the video sequence. We use a skin detection algorithm developed by [20] that calculates the most likely skin pixels based on a previously computed skin model. Their skin models were trained using manually annotated skin pixels (14,985,845 pixels) and non-skin pixels (304,844,751 pixels) to form non-parametric histogram-based models. Once the skin likelihood values are calculated for each pixel in the video frame, we apply a chosen threshold (cur-

rently 0) and assign all pixels greater than the threshold as skin pixels. Next, we perform a morphological close operation on the image to remove single pixels and to form a face blob in the image. Finally, we calculate the bounding box around the face blob and use the bounding box to extract the face portion of the original video frame.

Once we have located the face portion of the video frame, we can extract the face, align it, and normalize it to a template face image. Let us assume a face patch I, a normalized frontal template face image T, a warping $W(x, p)$, in which x is the image coordinates and p is the set of affine similarity transformation parameters. Also let r denote the patch index. To find the best warping, we seek to minimize the following error function with respect to $\Delta\mathbf{p}$:

$$E_r = \sum_x [I_r(W(x, p + \Delta\mathbf{p})) - T_r(x)]^2 \tag{3.1}$$

Minimizing E_r is a non-linear optimization task. To solve it linearly, we use the Lucas-Kanade image alignment algorithm [60]. In short, the solution is

$$\Delta\mathbf{p} \;=\; H^{-1} \sum_x (\nabla I_r \frac{\partial W}{\partial p})^T (T_r(x) - I_r(W(x, p))) \tag{3.2}$$

where $\Delta\mathbf{p}$ are the parameter updates and H is the Hessian matrix given by

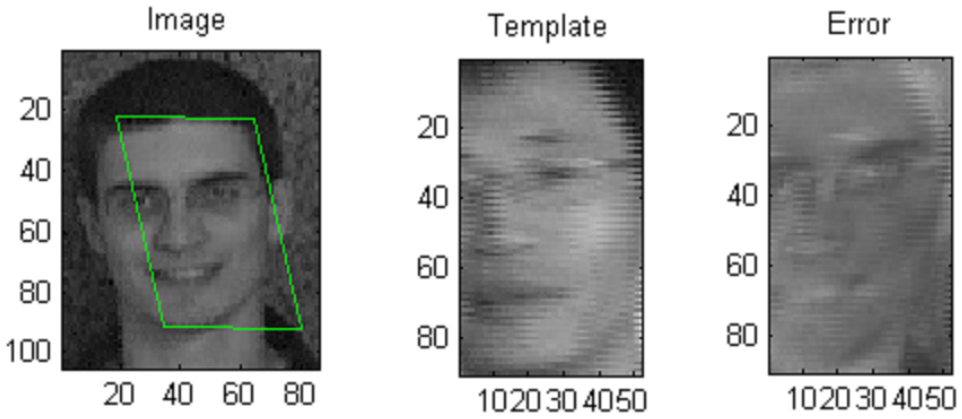


Figure 3.2: Patch alignment

$$H = \sum_{x} (\nabla I_r \frac{\partial W}{\partial p})^T (\nabla I_r \frac{\partial W}{\partial p}). \tag{3.3}$$

The alignment algorithm iterates until $\Delta \mathbf{p}$ is sufficiently small. Figure 3.2 illustrates a patch alignment example from our video dataset. Since the template has a slightly different appearance from that of each individual patch, this step cannot locate the patch to a precise position on the template. Therefore, a refinement step is required to align the patches more accurately.

3.2.2 Face Patch Stitching

In the previous step, each patch is warped to roughly the correct location and pose. To construct a full face at the pixel level precision, we develop an image stitching algorithm. We wish to minimize the overlapping error be-

tween the patches. The set $\mathbb{S}$ of n warped frontal face patches is denoted $\mathbb{S} = \{J_1, J_2...J_n\}$. Our goal is to find the optimal alignment of the set of face patches by minimizing the following error function:

$$E_{\mathbb{S}} = \sum_{\Omega} (J_i(W(x, p_i)) - J_k(W(x, p_k)))^2 \tag{3.4}$$

to produce the reconstructed face J', where i and k ($i \neq k$) correspond to different patches and Ω is the overlapping region between aligned patches. To solve this equation, the algorithm loops between each patch pairs and iterates to refine the parameters.

Post-processing is performed on the reconstructed face J' to improve face recognition results. The pixels on the overlapping regions are taken from the average values of each of the contributing regions. The boundary pixels of the patches are Gaussian smoothed by local 3×3 windows to eliminate the patch line artifacts. Figure 3.3 shows an example of the patch stitching step on a set of face patches from video.

3.3 Recognition from the Reconstructed Face

The final task for our methodology is to classify the reconstructed face to the most likely candidate face in our training data. Because we are reconstructing the face using patches from video, it is likely that the reconstructed face will have missing data. Face recognition via sparse representation introduced by Wright *et al.* [85] is employed to handle the recognition task with

Figure 3.3: Reconstruction example. Upper row: set $\mathbb{S}$ of face patches; Lower row: reconstructed face image J'.

missing data. Section 2.1.3.2 introduces the sparse representation classification method, so we will briefly summarize the method here.

Essentially, an optimization problem is solved to represent a test image u by a linear combination of all training images. If we restrict the solution to be sparse, then find the most sparse solution to $Ax = u$, which leads us to the classification of our test image by matching it with the subject with the most non-zero entries in x. The real power of this method is that it is robust to occlusions in the test image. This is due to the assumption that an occlusion

can be modeled as a sparse error that affects only a few pixels in the image. We write the equation $y = Ax + Ie$, where y is the test image, A is the set of training vectors, x is the set of sparse coefficients, I is the identity matrix and e is the error vector due to occlusion. Then, we can rewrite the equation as $y = Cf$, where $C = [AI]$ and f captures both the sparse coefficients and the error vector. Now, if the $L1$ minimization is performed on f, the sparsity constraints will be imposed on both x and e, which will lead to a solution that is robust to occlusion. Of course, far too much occlusion can degrade the performance. For more details, please refer to [85].

3.4 Experiments

In this section, we present our two experiments to test the proposed algorithms. In the first experiment, we use the well known Yale Face database [8]. The face patches are generated from chosen locations in the original image and are used to reconstruct a full face from the patches. Four face patches per image are generated by randomly adding pixel noise to the known location of the patch with variances of 5, 10, 15 and 20 pixels. Figure 3.4 shows the average pixel error between the reconstructed faces and the original faces. The mean pixel error for all images was 16.9 for variance of 5 pixels and 22.4 for variance of 20 pixels with a standard deviation of 3.9 and 4.8 pixels respectively. We then split the database in half; the first half of the original images are used for training and the second half of the reconstructed images are used for testing. Then we employ face recognition via sparse representation to recognize the

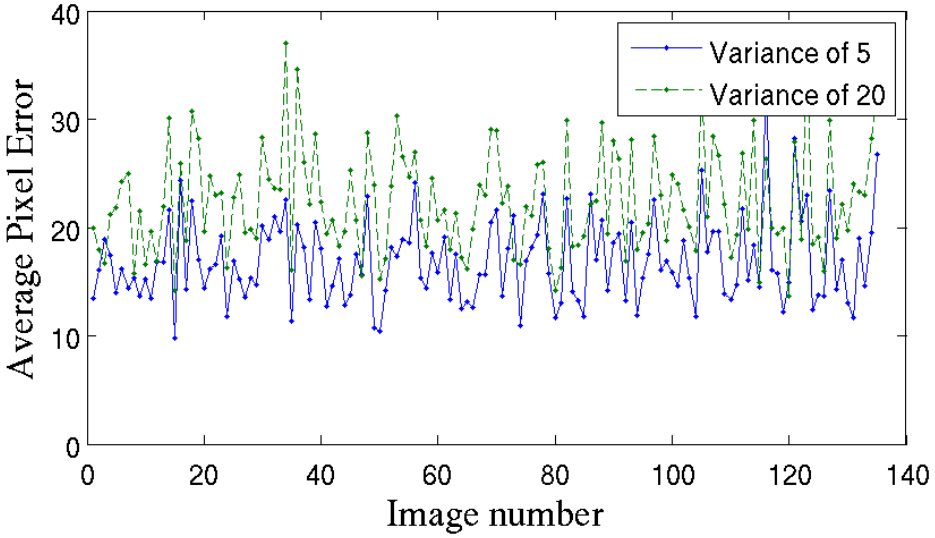


Figure 3.4: Yaleface face reconstruction error

reconstructed faces. Figure 3.5 compares the recognition rates of the original images and the reconstructed faces. The recognition rate is the same for low variance and only slightly lower when variance is increased.

In the second experiment, we test the algorithm using a video sequence generated in our lab. In the video sequence, both full faces and part faces of each of the 7 subjects are present. We use only face patches for reconstruction. We compare the recognition rates of the reconstructed face images to that of the full face examples from the video. With two full face images for training and six reconstructed face images per subject, we were able to recognize 34 out of 42 of the reconstructed faces correctly for an accuracy rate of 81%. For comparison, when using one full face image per subject for training and one

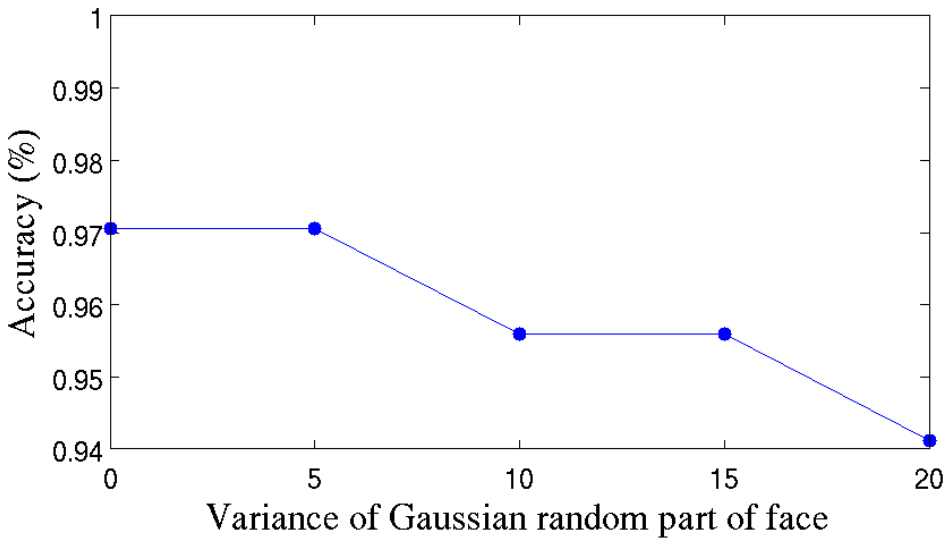


Figure 3.5: Yaleface database recognition rate comparison

per subject for testing, all test images are recognized correctly. Figure 3.6 shows four of the full and reconstructed images from the video sequence.

3.5 Conclusion

In this chapter, a novel method for video-based face recognition is proposed. We collect face patches from video and stitch them to reconstruct a still face image. Our methodology uses face recognition via sparse representation to recognize reconstructed faces. Sparse representation can handle noise and occlusion better than other algorithms such as PCA and ICA. Because our reconstructed face can come from patches of different views, self occlusion and region rectification errors can introduce severe noise in the reconstructed

Figure 3.6: Training and testing examples for video-based face recognition. Upper row: Training faces. Lower row: Reconstructed faces from video.

image. Sparse representation is an effective tool for this task. Our experiments show that this method reaches a high recognition rate considering that there is missing data in the reconstructed face. This method helps to transform the video-based face recognition problem to the still face recognition problem, which enables the application of still face recognition algorithms in video face recognition. The patch-based method does not need a complex face model, such as a 3D or cylinder head model. It is flexible and more general than other methods. The limitation of this method is that large changes in pose, illumination and expression cannot currently be handled, which will be addressed in future work. Another extension of this work is to use the redundant information present in the overlapping patches. The redundancy could help to eliminate noise and produce a higher quality image. Utilizing symmetry, one

type of redundancy, could produce an improved recognition rate, which is the case in [34].

Chapter 4

Fusing Face Recognition from Multiple Cameras

In this section, we introduce our method for fusing face recognition results from multiple video cameras. By fusing the results from multiple cameras, the overall face recognition accuracy may be improved. Figure 4.1 depicts an overview of the system. The system consists of two main parts. First, the face is tracked in a video sequence from multiple cameras using a cylinder head model which is used to extract face texture from the video frames. Second, the results from face recognition of the extracted face texture are fused from each of the cameras to improve the overall face recognition accuracy. The following pages describe our system in detail.

4.1 Face Tracking with Cylinder Head Models

In order to translate the problem of face recognition from video to a still face recognition problem, we desire a method to robustly track the face of an individual from multiple cameras so that we may combine the tracking results in a meaningful way. The cylinder head model (CHM) [86] has several advantages. First, CHMs are able to recover the full-motion parameters (3

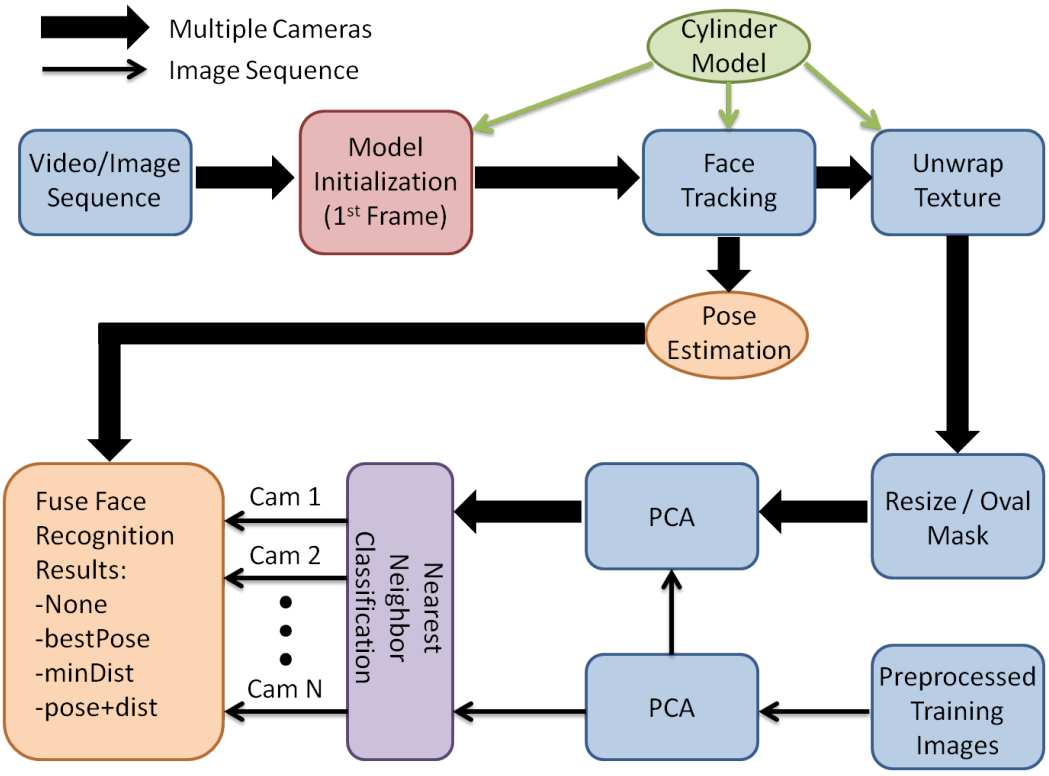


Figure 4.1: Overview of Fusing Face Recognition Results from Multiple Cameras.

rotations and 3 translations) of the head. Since this work deals with multiple surveillance cameras, the recovery of these parameters is crucial in order to fuse information about the head and face in all camera views. A summary of the cylinder head model and tracking algorithm found in [86] follows.

The cylinder head model makes a basic assumption that we can treat the head (and thus face) as a cylinder. Therefore, rotation or translation performed by the head can be estimated by a cylinder with 6 parameters (3 rotations and 3 translations). Let the vector $\boldsymbol{\mu}$ represent the rigid motion, including 3D rotation $(\theta_x,\ \theta_y,\ \theta_z)$ and the translation $(t_x,\ t_y,\ t_z)$. If $\boldsymbol{x} = (x, y, z)^T$ is a 3D coordinate of a point on the cylinder surface, then the new location of $\boldsymbol{x}$ after rigid motion transformation by $\boldsymbol{\mu}$ is

$$\boldsymbol{M}(\boldsymbol{x}, \boldsymbol{\mu}) = \boldsymbol{R}\boldsymbol{x} + \boldsymbol{T}, \tag{4.1}$$

where $\boldsymbol{M}$ is the function of the rigid transformation, $\boldsymbol{R}$ is the rotation matrix and $\boldsymbol{T}$ is the translation vector. The rigid motion of the head from time t to time $t+1$ is described by the change in the rigid motion vector, $\Delta\boldsymbol{\mu}$. Therefore, if $\boldsymbol{p}_t = (u, v)$ is the projection point in the image plane $\boldsymbol{I_t}$ of point $\boldsymbol{x}$ on the cylinder in 3D (which are depicted in Figure 4.2), then the new location of point $\boldsymbol{p}_{t+1}$ in the next frame $\boldsymbol{I}_{t+1}$ is estimated as

$$\boldsymbol{p}_{t+1} = \boldsymbol{G}(\boldsymbol{p}_t, \Delta\boldsymbol{\mu}) \tag{4.2}$$

and the next frame can be computed by

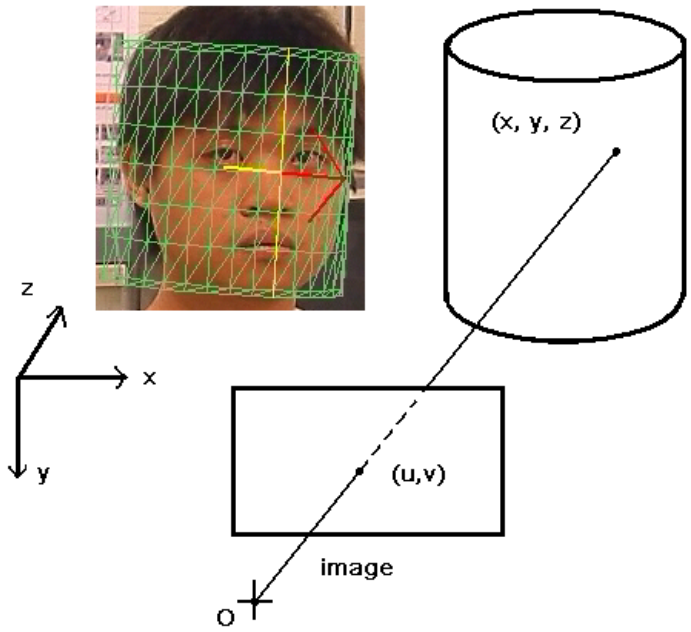


Figure 4.2: Relationship between points on the 3D cylinder model and the image plane.

$$\boldsymbol{I}_{t+1}(\boldsymbol{G}(\boldsymbol{p_t}, \Delta\boldsymbol{\mu})) = \boldsymbol{I}_t(\boldsymbol{p}_t), \tag{4.3}$$

where $\boldsymbol{G}$ is the 2D parametric motion function of $\boldsymbol{p_t}$. In this estimation we assume that the illumination does not change between frames, so the pixel intensities between the two frames are consistent.

The change in rigid motion vector $\Delta\boldsymbol{\mu}$ can be obtained through a minimization of the error between two successive image frames and can be solved by using the Lucas-Kanade image alignment algorithm [60]. The solution is

$$\Delta\boldsymbol{\mu} = -(\sum_{\Omega}(\boldsymbol{I_p G_\mu})^T(\boldsymbol{I_p G_\mu}))^{-1}\sum_{\Omega}(\boldsymbol{I_t}(\boldsymbol{I_p G_\mu})^T) \tag{4.4}$$

where $\boldsymbol{\Omega}$ is the region of overlapping pixels between the two frames, $\boldsymbol{G_\mu}$ is the partial derivative of $\boldsymbol{G}$ with respect to the rigid motion vector, and $\boldsymbol{I_p}$ and $\boldsymbol{I_t}$ are the spatial and temporal image gradients, respectively.

Assuming that the camera projection matrix depends only on the focal length, the derivative of $\boldsymbol{G}$ with respect to the rigid motion vector at $\boldsymbol{\mu} = 0$ is [86]

$$\begin{aligned}\boldsymbol{G_\mu}\Big|_{\Delta\boldsymbol{\mu}=0} \\ = \begin{bmatrix} -xy & x^2+z^2 & -yz & z & 0 & -x \\ -(y^2+z^2) & xy & xz & 0 & z & -y \end{bmatrix}\frac{f}{z^2},\end{aligned} \tag{4.5}$$

where f is the focal length of the camera and x, y and z are the 3D coordinates.

By plugging the result of (4.5) into equation (5.5), the rigid head motion vector $\Delta\boldsymbol{\mu}$ is recovered.

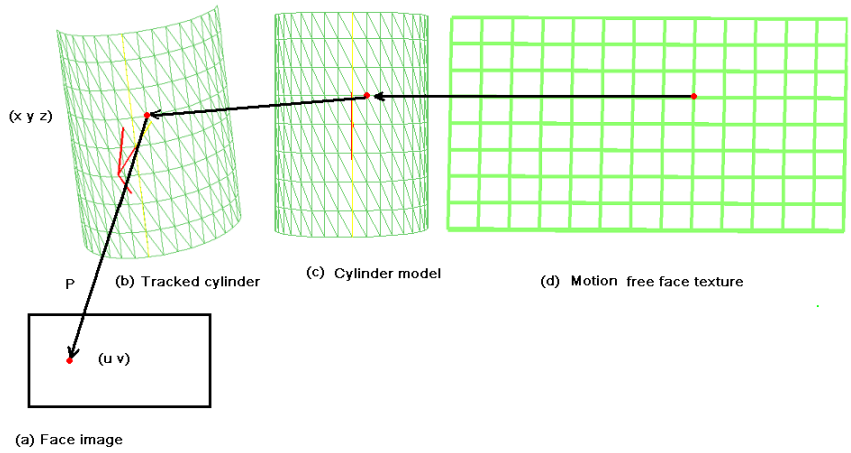


Figure 4.3: Cylinder Tracking Result.

Using the CHM tracking result, we scan the image to the cylinder, then unwrap the cylinder to a standard texture map as portrayed in Figure 4.3. The pixel value for each point in the texture image ((d) in Figure 4.3) is found by locating the point on the cylinder model ((c) in Figure 4.3), finding the corresponding location on the tracked cylinder ((b) in Figure 4.3) and finally estimating the value of the pixel from the original face image ((a) in Figure 4.3). In the ideal case of perfect tracking, the texture map is stabilized from global motion and produces a frontal face that is centered horizontally in the unwrapped image.

An example tracking result using the CHM is shown in Figure 4.4 in which a single person is tracked from two cameras. Each row of the figure

represents a single frame from both cameras. The images in Figures 4.4(a), 4.4(e), 4.4(i) and 4.4(m) are from camera A while the images in Figures 4.4(d), 4.4(h), 4.4(l) and 4.4(p) are from camera B. The image pairs in the center of Figure 4.4 are the unwrapped cylinder images from their corresponding CHM in the original images.

4.2 Fusing Face Recognition from Multiple Cameras

Incorporating the results of multiple cameras viewing a common subject may increase the accuracy and robustness of the face recognition task. In this work, the face recognition results of multiple cameras are fused. Since eigenfaces is used along with NN for the face recognition task, a distance between the projected testing weights and the projected training weights is calculated. Let us define camera A as the camera that views mostly the right half of the face and camera B as the camera that views mostly the left half of the face. Considering the case of a two-camera system, at every time t, there will be a frame from camera A that corresponds to a frame from camera B. Therefore, each frame will have a minimum distance calculation that will be used to assign the classification result of the face in each frame. Also, by using the CHMs to track the face in each frame, an estimate of the pose of the face (only yaw in our case) is calculated. These two pieces of information (distance to classified training sample and pose estimation) are used in the combination of results from multiple cameras. We present results using 5 different methods for fusing the results between the two cameras:

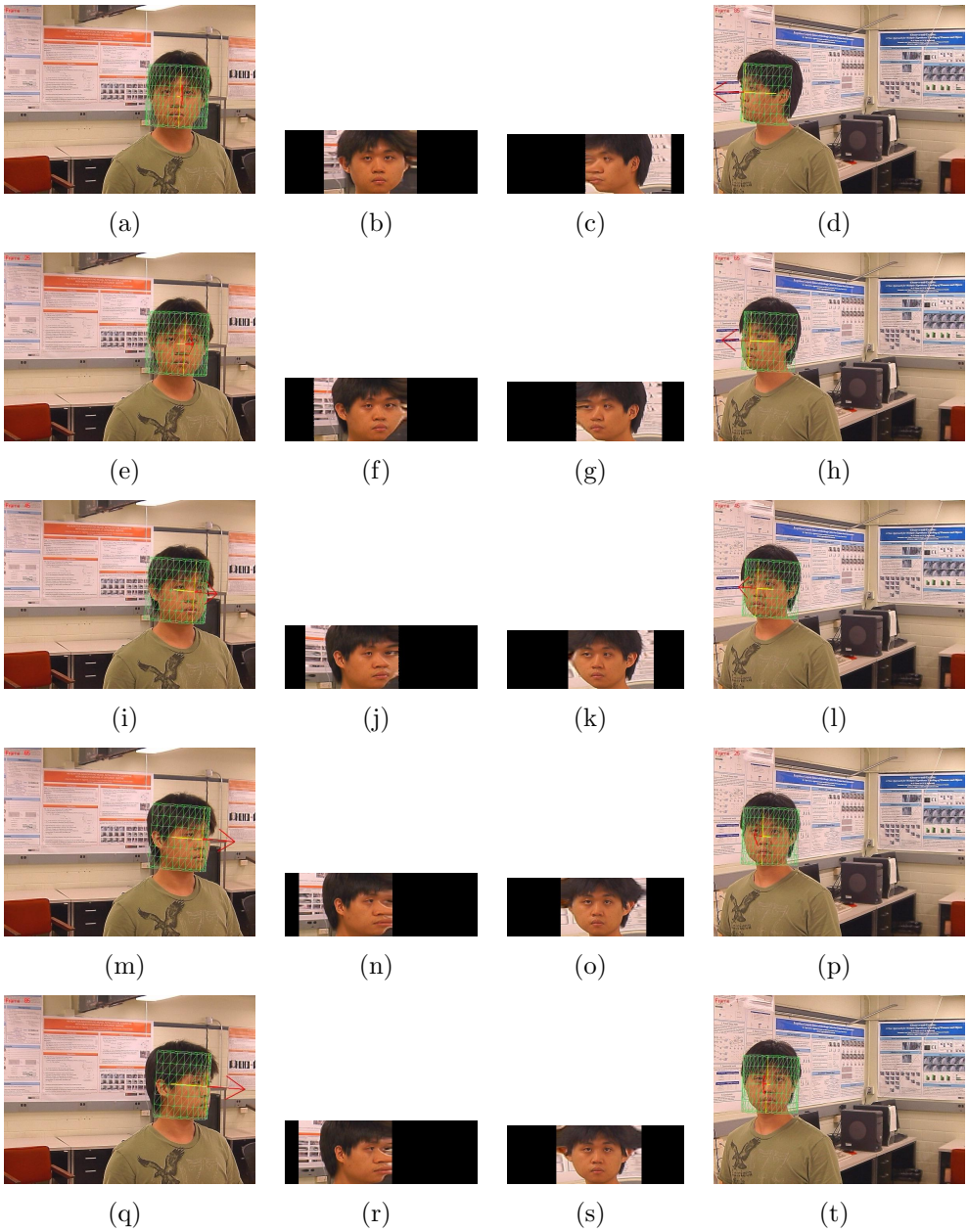


Figure 4.4: CHM Tracking and Scanning Result on 5 Pairs of Images from the Two Cameras.

1. Independent (Ind). Each face is recognized independently as if it was from a single camera, so no combination of the two cameras is used.

2. Minimum distance (MinDist). Between the two cameras, the camera with the minimum NN distance is chosen as the classification result.

3. Best pose (BestPose). The camera with the most frontal pose is used for the classification result.

4. Multiplier weights (MultWts). The NN distance and pose information of the two cameras are multiplied along with a constant and the minimum of this new distance is used for the classification result.

5. Gaussian weights (GaussWts). The pose of each of the cameras is used to produce a Gaussian weight which is then applied to the NN result. The minimum of this result is used for recognition.

The distance from the testing image weights to the nearest training image weights gives some measure of how close the two samples are in "face space". The estimated pose of the face gives a measure of the ability to recognize the subject's face correctly. For instance, a full frontal face should be easier to classify than a non-frontal face with a yaw of 30° or more. Each method of fusing the results requires a straightforward calculation that may have dramatic accuracy gains to the multi-camera face recognition system as discussed in the experiments section.

A brief description of each of these fusion methods is presented.

4.2.1 Independent

This result assumes that there is no information to share between the two cameras. Each face is recognized on its own independent of pose or camera location. This is used as a baseline for the other results.

4.2.2 Minimum Distance

Since a NN distance is calculated for each of the cameras, the simplest way to fuse the results from the cameras using only the NN distances is to choose the minimum distance for classification. Therefore, the label of the training image of the camera with the minimum NN distance is chosen for the recognition result.

4.2.3 Best Pose

Using only the calculated pose information of each of the cameras from the CHM model, we can fuse the recognition results by simply choosing the camera with the most frontal pose. Therefore, the fused recognition result is the label applied to the most frontal face image of the two cameras.

4.2.4 Multiplier Weights

The first attempt to combine both the pose information and the NN distance from each of the cameras is the simplest. We form a normalized pose from the pose calculation of cameras A and B (P_A and P_B, respectively) by dividing it by the maximum pose ($Pnorm_A = \frac{|P_A|}{max(P)}$ and $Pnorm_B = \frac{|P_B|}{max(P)}$).

Then we multiply the NN distances from each of the cameras by the normalized poses and use the minimum result for classification.

4.2.5 Gaussian Weights

In the final fusing of recognition results between the two cameras, a Gaussian weighted approach is used. Let d_A be the Euclidean distance between the projected weights from the face tracked in camera A to the nearest training sample's projected weights and let d_B be similarly defined. Let P_A and P_B be the estimated pose of the face (where a frontal face image has a yaw of 0°) from the CHM calculated from the frames in camera A and B respectively. Weights w_A and w_B are calculated for each pair of frames using the pose estimations and a Gaussian function as

$$\begin{aligned} w_A &= 1 - \frac{1}{\sigma\sqrt{2\pi}} e^{-(Pnorm_A)^2/2\sigma^2} \\ w_B &= 1 - \frac{1}{\sigma\sqrt{2\pi}} e^{-(Pnorm_B)^2/2\sigma^2} \end{aligned} \tag{4.6}$$

where $Pnorm_A = \frac{|P_A|}{max(P)}$ and $Pnorm_B = \frac{|P_B|}{max(P)}$) are the ratios of estimated poses to the maximum pose, the mean of the Gaussian was chosen to be zero (frontal pose) and $\sigma = 0.01$, which favors frontal poses and penalizes poses that are non-frontal. The maximum pose ($max(P)$) calculated from our video was 70°. The calculated weights are then multiplied to produce new distance measures D_A and D_B by

$$D_A = d_A * w_A$$
$$D_B = d_B * w_B. \tag{4.7}$$

The resulting modified distances are then used for classification. If $D_A \leq D_B$, the training label applied to the face from camera A is applied to the face in the camera A and camera B pair. Otherwise the label from camera B is chosen.

4.3 Building Confidence by Aggregating Results

One contribution we have made is to use face recognition results from several frames of a successfully tracked subject to give a better recognition result with a certain level of confidence. In the videos used in this work, each video sequence has only one subject, so the successful tracking of the face is made simpler. However, one could imagine a scenario in which several subjects are tracked in the same video sequence and the tracking results of each subjects' faces are successful. The results of the face recognition on each of the pair of frames of a single subject's tracking result using the fused results method explained in Section 4.2 are aggregated together. This aggregation of results is a simple scoring process. For instance, if 100 pairs of frames are labeled as subject 2 and 50 pairs of frames are labeled as subject 1 in a 150 frame tracking sequence, we would label the subject of the track as subject 2 with a confidence of 100/150, or 66.7%. This method gives intuitively positive results based on the assumption that the more frames you have of an

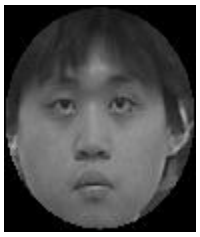


Figure 4.5: Result of Centered and Masked Face Image

individual, the more confidently the correct identification will be applied.

4.4 Still Face Recognition Method

An oval mask is first applied to a face image that has been successfully tracked by the CHM so that background noise is removed for the recognition process. One example of the application of this mask to the face image is found in Figure 4.5. Once the mask has been applied to every training and testing image in the database, recognition of the faces can proceed.

4.4.1 Eigenfaces

Eigenfaces [79] is used for recognition for mainly two reasons. We are interested in transforming the problem of face recognition from video to a still face recognition problem so that any still face recognition algorithm can be used. Also, the implementation of the eigenfaces algorithm is well studied and needs little discussion.

Nearest-neighbors (NN) is used to classify the test weights to the nearest training sample using the Euclidean distance between the weights.

4.5 Experimental Results

Video sequences taken from two cameras with unique views of the face are used for the experiments. Each subject varied the pose of their face in each of the video sequences by changing the yaw (or pan) from 0° to ±70°. Example images from our video sequences are shown in Figure 4.4. This variation in pose produced 100 frames per subject (for a total of 6 subjects) from each of the cameras which were used as test data. Two frontal images per subject are used for the training data.

Two face recognition experiments are performed on the two-camera video sequences. In both experiments, the face in the video is tracked using CHMs and the estimated pose is returned for each frame. Eigenfaces is applied to the cropped faces tracked in each frame directly in the first experiment. This is used in comparison to our methodology in the second experiment of using an unwrapped cylinder face image. The first 12 eigenfaces of the training data is used as the low-dimensional feature space for classification. The Euclidean distance was used as the distance measure for nearest-neighbor (NN) classification. These two experiments are used along with the 5 methods for fusing the recognition results described in Section 4.2 for a total of 10 results (5 methods times 2 experiments).

The results of the experiments are displayed in Table 4.1. The results from the original 2D cropped face images from the tracking results are displayed under the "Orig" column, while the results gathered by using the faces generated by the CHM tracking and cylinder unwrapping are displayed un-

Table 4.1: Face Recognition Results.

	Orig	CHM
Ind	72.6%	67.4%
MinDist	82.2%	94.4%
BestPose	76.2%	91.5%
MultWts	75.9%	93.5%
GaussWts	81.9%	94.4%

der the "CHM" column. The "Ind", "MinDist", "BestPose", "MultWts", and "GaussWts" results are those generated by fusing the recognition results from each of the cameras by using the NN distance and/or pose information as described in Section 4.2. The highest accuracy reported is 94.4% by using the images generated by the CHMs and the fused results from both cameras with methods "MinDist" and "GaussWts".

As mentioned in Section 4.3, one can use multiple frames from a successfully tracked face to build confidence in the recognition result. In the experiment using the CHM tracked faces and the fused results (using the GaussWts method), we achieve 100% accuracy for all subjects with an average confidence of 93.8%.

4.6 Discussion

Clearly, the idea of fusing the recognition results from both of the cameras is more successful than independently recognizing the results. This fusing method alone is responsible for an increase in accuracy of 4% to 25% in our experiments. Using CHMs to track the face and produce an unwrapped cylin-

der face image for recognition has be shown to outperform the original 2D face captured from the video frames by almost 20%, but only in the case of fusing the recognition results. However, to achieve the recognition result of 94.4%, both methods were necessary. The two most successful methods for fusing the recognition results between the two cameras appear to be the minimum distance and Gaussian weights methods. We suspect that with a larger database, the minimum distance method alone would tend to produce a lower accuracy than using the Gaussian weighted method which uses pose information, but this needs to be tested. The further step of aggregating recognition results of a subject that has been successfully tracked in video produces 100% recognition on the all of the subjects tracked in our video sequences with most confidence levels above 90%. It is possible that with a faster frame rate and/or longer video sequences, the confidence of the recognition results could be further improved.

4.7 Conclusion

Face recognition from video presents many challenges such as self-occlusion, occlusion from objects, and illumination changes. We present a method to overcome the problems of self-occlusion and lack of frontal face images in video by using CHMs to produce an unwrapped cylinder face image and the estimated pose of the face. Using these outputs we fuse the face recognition results of both cameras, which results in a dramatic increase in accuracy. Eigenfaces is used for the face recognition task. The proposed method

achieves an accuracy of 94.4%. By further aggregating recognition results from a successfully tracked individual, a recognition rate of 100% is achieved with high confidence.

Future work includes improving the CHM tracking so that initialization using a frontal face is avoided as well as improving the quality and pose of the unwrapped cylinder image. Perhaps the use of a 3D mesh model would improve the transformation from a non-frontal pose to a frontal face image, however, this could require more computation. Generalizing the method to other face recognition algorithms is desired. One could also combine the unwrapped images from the CHMs in each camera in a more direct manner to reconstruct a full face image for recognition.

Chapter 5

Robust Multiple Camera Face Tracking

In order to obtain accurate face recognition results from a video-based face recognition system, the face must be robustly tracked in as many frames of the video as possible. To increase the robustness of face tracking, we introduce a method for full-motion recovery of the face from multiple video cameras. Figure 5.1 displays an outline of our method. While it may seem obvious that introducing multiple cameras with overlapping views of the face may improve the face tracking result, it is not obvious how to combine the information present in the cameras. We present a novel method which jointly estimates the motion of the face, which is outlined in Figure 5.2.

We demonstrate the proposed approach using video sequences with ground truth of 3D head motion from real data. Comparing the motion recovery from multiple cameras with that of a single camera, we show that the accuracy of pose estimation has been significantly increased. We also find that the motion-free texture of the face generated from the cylinder model with the multiple camera tracking produces higher recognition rates compared with the single camera case.

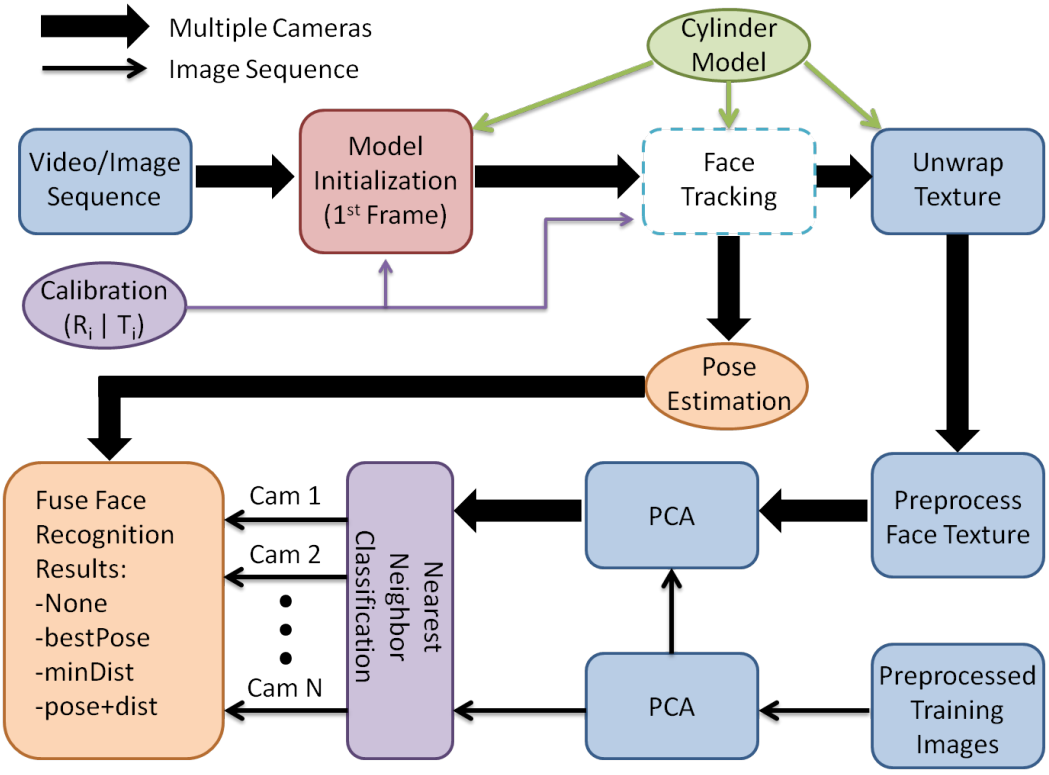


Figure 5.1: Overview of our Multiple Camera Face Tracking Method

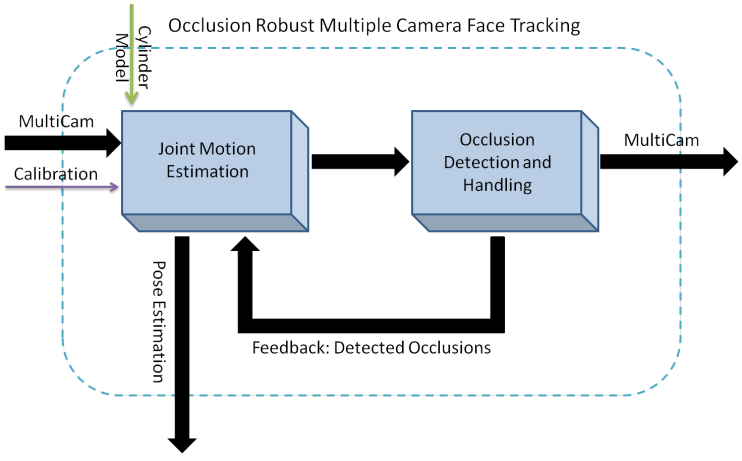


Figure 5.2: Occlusion Robust Face Tracking

5.1 Background

As we have previously stated, the task of accurately tracking the face in a video sequence is crucial to the ability to recognize the individual. Before we introduce our multiple camera tracking method, We will briefly introduce related work in the areas of face tracking and pose estimation.

5.1.1 Face Tracking and Pose Estimation

One important class of object tracking research is face (or head) tracking. Most face recognition and facial expression analysis methods require the motion of faces to be known so that faces may be aligned, implicitly or explicitly. One application of face (and object) tracking is pose estimation, which is a very large and diverse research topic, as depicted in [63]. These methods may be classified as feature-based approaches or model-based approaches. However, only a small portion of these algorithms recover the 3D face motion. The authors in [45, 57, 58] track image features to recover 3D poses. Feature-based approaches are flexible but performance depends heavily on the availability of good features. Since human heads are similarly shaped, many model-based approaches have been proposed to take advantage of this similarity. In [2, 18, 48, 76, 86], the head is treated as a cylinder and the head motion is recovered using a cylinder model. Some researchers apply ellipsoid models instead [6, 19, 26]. Model-based approaches are more reliable considering that the motion recovery is obtained from the whole face region. Yet, the whole face region must be visible to perform the motion calculation. Therefore, large

nonfrontal poses of the face still challenge the effectiveness of the model-based method. However, by utilizing multiple surveillance cameras with overlapping views, we may improve the model-based approach by fusing the motion from the multiple views into a joint 3D motion estimation.

Previous researchers in multi-camera face tracking have used stereo information to increase tracking robustness, such as in [46]. In contrast to stereo, we desire the integration of multiple camera information to obtain as many observations as possible to recover the head motion. The robustness and accuracy of head motion tracking may be increased by integrating the motion information from multiple cameras. Our work is most similar to that of Cai *et al.* [13], where they propose a method that integrates feature tracking of the face from multiple cameras by adapting a generic head model. In addition to using a more generic model that is easier to initialize, our method also produces texture of the face that is appropriate for face recognition. Feature-based tracking relies on features of the face being present in every frame, whereas our use of model-based tracking does not have this limitation. Additionally, an individual camera may recover the pose correctly if the tracking is lost in that camera's view with assistance from other cameras in a multiple camera setting. However, how does one combine the motion from multiple cameras effectively to improve the face tracking result? We present a complete derivation for explicitly fusing the motion from multiple cameras into a joint 3D (three rotations and three translations) motion estimation of the face.

5.2 Full-Motion Recovery from Multiple Cameras

In this section we introduce the multiple camera full-motion recovery model. Without loss of generality, we specify one of the cameras as the world coordinates for our system. We call this camera the "first" camera. This is done because the derivation for the first camera motion is unique from the derivation of the other cameras in the system, since the motion of the 3D rigid object will be calculated in the first camera's view. Incidentally, the derivations for a single camera motion and the first camera in the multiple camera motion are essentially the same. The derivations for the first camera and the extension to multiple cameras follows.

5.2.1 First Camera Motion

In a model-based head tracking approach, a basic assumption is made that the head (and thus the face) may be treated as a 3D rigid object. Therefore, only six parameters (three translations and three rotations) are needed to describe the motion performed by the head. The motion ($\boldsymbol{M}$) of the 3D points on the rigid object w.r.t. the first camera coordinates may then be described as

$$\boldsymbol{X}(t+1) = \boldsymbol{M} * \boldsymbol{X}(t), \tag{5.1}$$

$$\boldsymbol{M}(\boldsymbol{x}, \boldsymbol{\mu}) = \boldsymbol{R}\boldsymbol{x} + \boldsymbol{T}, \tag{5.2}$$

where $\boldsymbol{X}(t+1)$ are the new coordinates of the 3D points $\boldsymbol{X}(t)$ after a motion of $\boldsymbol{M}$ has been applied where $\boldsymbol{\mu}$ is a six element vector representing rigid motion, including 3D rotation (w_x, w_y, w_z) and translation (t_x, t_y, t_z), $\boldsymbol{x} = (x, y, z)^T$ is a 3D coordinate of a point on the surface of the object, $\boldsymbol{M}$ is the function of the rigid transformation, $\boldsymbol{R}$ is the rotation matrix and $\boldsymbol{T}$ is the translation vector. We denote the rigid motion of the head from time t to time $t+1$ as $\Delta\boldsymbol{\mu}$. If $\boldsymbol{p}_t = (u, v)$ is the projection point in the image plane $\boldsymbol{I_t}$ of point $\boldsymbol{x}$ on the 3D object , then the new location of point $\boldsymbol{p}_{t+1}$ in the next frame $\boldsymbol{I}_{t+1}$ is estimated as

$$\boldsymbol{p}_{t+1} = \boldsymbol{F}(\boldsymbol{p}_t, \Delta\boldsymbol{\mu}). \tag{5.3}$$

The next image frame may then be computed by

$$\boldsymbol{I}_{t+1}(\boldsymbol{F}(\boldsymbol{p_t}, \Delta\boldsymbol{\mu})) = \boldsymbol{I}_t(\boldsymbol{p}_t), \tag{5.4}$$

where $\boldsymbol{F}$ is the 2D parametric motion function of $\boldsymbol{p_t}$. A necessary and reasonable assumption is made that the illumination does not change and that movement is small between frames, so the pixel intensities between the two frames are consistent.

To compute the change in rigid motion vector $\Delta\boldsymbol{\mu}$, the error between two successive image frames is minimized. This is solved by using the Lucas-Kanade image alignment algorithm [60]. The result is

$$\Delta\boldsymbol{\mu} = -(\sum_{\Omega} \boldsymbol{G}^T\boldsymbol{G})^{-1} \sum_{\Omega} \boldsymbol{I_t}\boldsymbol{G}^T \tag{5.5}$$

where

$$\boldsymbol{G} = \boldsymbol{I_p}\boldsymbol{F_\mu} \tag{5.6}$$

and where $\boldsymbol{\Omega}$ is the region of overlapping pixels between the two frames, $\boldsymbol{F_\mu}$ is the partial derivative of $\boldsymbol{F}$ w.r.t. the rigid motion vector, and $\boldsymbol{I_p}$ and $\boldsymbol{I_t}$ are the spatial and temporal image gradients, respectively.

Then, under the assumption that the perspective projection only depends on the focal length, then the derivative of $\boldsymbol{F}$ w.r.t. the rigid motion vector at $\boldsymbol{\mu} = 0$ is [86]

$$\boldsymbol{F_\mu}\Big|_{\Delta\boldsymbol{\mu}=0} = \begin{bmatrix} -xy & x^2+z^2 & -yz & z & 0 & -x \\ -(y^2+z^2) & xy & xz & 0 & z & -y \end{bmatrix} \frac{f}{z^2}, \tag{5.7}$$

where x, y and z are the 3D coordinates of the object and f is the focal length of the camera. For single camera tracking, the rigid head motion vector $\Delta\boldsymbol{\mu}$ is recovered by substituting the result of (5.7) into equation (5.6). Then, using Rodrigues' transformation formula, $\boldsymbol{M}$ is calculated from $\Delta\boldsymbol{\mu}$ and applied in equation (5.1) to recover $\boldsymbol{X}$ at time $t+1$.

5.2.2 Multiple Camera Motion

The most natural way to extend the full-motion recovery model to multiple cameras is to allow each camera to track the face independently. However, if any of the independent cameras lose track of the face due to large nonfrontal poses of the face, it may not be able recover the track and pose of the face. Also, by combining the motion information from multiple cameras simultaneously, we may improve the robustness of the overall motion estimation of the face. Since each of the cameras are viewing the face of the same individual, the 3D motion of the face must be the same w.r.t. the world coordinates. Therefore, the motion that is described in each of the cameras may be used to estimate the motion of the face more accurately and precisely. We take advantage of this observation by calculating a joint change in motion from the cameras.

Using a similar notation and methodology found in the first camera section, we recover the full motion of the face from multiple cameras in the following manner. Please note that the following derivation could be applied to any number of cameras that refer back to the first camera as the world coordinate system. In the first camera's view, we have

$$\boldsymbol{p}_1(t+1) = \boldsymbol{K}_1 * \boldsymbol{X}_1(t+1) = \boldsymbol{K}_1 * \boldsymbol{M} * \boldsymbol{X}_1(t) \tag{5.8}$$

where $\boldsymbol{p}_1(t+1)$ is the projection of those 3D points to the first camera's image plane obtained through the multiplication with intrinsic camera matrix $\boldsymbol{K}_1$.

Recall that the motion of the 3D object in the first camera's view is shown in equation (5.1). To relate the *ith* camera with the first camera,

$$\boldsymbol{X}_1(t) = \boldsymbol{C}_i * \boldsymbol{X}_i(t), \tag{5.9}$$

where

$$\boldsymbol{C}_i = \begin{bmatrix} r_{11} & r_{12} & r_{13} & \tau_x \\ r_{21} & r_{22} & r_{23} & \tau_y \\ r_{31} & r_{32} & r_{33} & \tau_z \\ 0 & 0 & 0 & 1 \end{bmatrix}_i \tag{5.10}$$

is the 4×4 matrix representing the rotation and translation between the *ith* camera's coordinate system and the world coordinate system (which is the first camera in our case), and $\boldsymbol{X}_i(t)$ are the 3D points w.r.t. the *ith* camera's coordinate system.

In the *ith* camera's view

$$\boldsymbol{X}_i(t+1) = \boldsymbol{C}_i^{-1} * \boldsymbol{X}_1(t+1) = \boldsymbol{C}_i^{-1} * \boldsymbol{M} * \boldsymbol{X}_1(t) \tag{5.11}$$

$$\boldsymbol{X}_i(t+1) = \boldsymbol{M}_i * \boldsymbol{X}_i(t) \tag{5.12}$$

$$\boldsymbol{p}_i(t+1) = \boldsymbol{K}_i * \boldsymbol{M}_i * \boldsymbol{X}_i(t) \tag{5.13}$$

where $\boldsymbol{p}_i(t)$ are the image coordinates in the *ith* camera's view after a projection with camera matrix $\boldsymbol{K}_i$. A simulated environment with three cameras viewing the motion ($\boldsymbol{M}$) of a cube is shown in Figure 5.3.

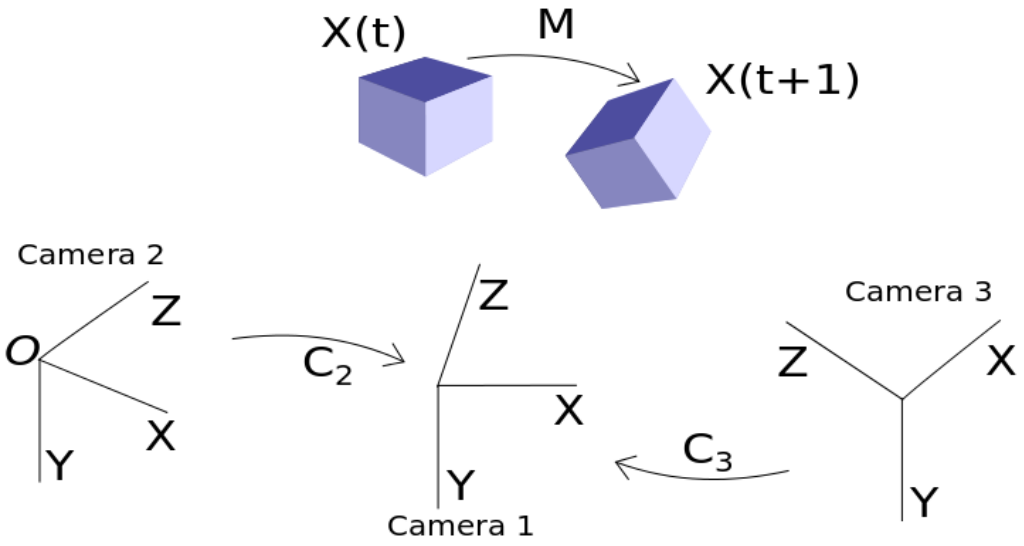


Figure 5.3: Example of a Three Camera System

Full motion recovery from multiple cameras is accomplished by relating the motion in each camera's view back to the motion in the first camera, $\boldsymbol{M}$. This may be done by using equations (5.11) and (5.12) in the following manner.

$$\boldsymbol{X}_i(t+1) = \boldsymbol{C}_i^{-1} * \boldsymbol{M} * \boldsymbol{X}_1(t) = \boldsymbol{M}_i * \boldsymbol{X}_i(t). \tag{5.14}$$

Therefore,

$$\boldsymbol{C}_i^{-1} * \boldsymbol{M} * \boldsymbol{X}_1(t) = \boldsymbol{M}_i * \boldsymbol{C}_i^{-1} * \boldsymbol{X}_1(t) \tag{5.15}$$

$$\boldsymbol{M}_i = \boldsymbol{C}_i^{-1} * \boldsymbol{M} * \boldsymbol{C}_i \tag{5.16}$$

and we may rewrite the equation for motion of the ith camera as

$$\boldsymbol{X}_i(t+1) = \boldsymbol{C}_i^{-1} * \boldsymbol{M} * \boldsymbol{C}_i * \boldsymbol{X}_i(t). \tag{5.17}$$

The reason the motion of the 3D points in the ith camera's view are represented this way is to exploit the idea that the motion of the 3D points is the same between multiple views of the moving object. We may now explicitly solve for the full-motion recovery of the face in both camera views and compute $\boldsymbol{\Delta\mu}$ from the information present in all cameras.

The crucial difference between calculating the motion in the ith camera's view as compared to the first camera's view is that the rotation and translation between the two cameras' coordinate systems (equation (5.9)) must be taken into account. Approximating the motion $\boldsymbol{M}$ by the twist representation [12] we have

$$\boldsymbol{M} = \begin{bmatrix} 1 & -w_z & w_y & t_x \\ w_z & 1 & -w_x & t_y \\ -w_y & w_x & 1 & t_z \\ 0 & 0 & 0 & 1 \end{bmatrix}, \tag{5.18}$$

which may also be thought of as a small angle approximation to the real rotation and translation. In the following equations, entries of the inverse matrix $\boldsymbol{C}_i^{-1}$ are denoted as r'_{jk} and τ'_j. Carrying through the projection in equation (5.13),

$$\boldsymbol{p}_i(t+1) = \frac{f_i}{z'} \begin{bmatrix} x' \\ y' \end{bmatrix} \tag{5.19}$$

where

$$
\begin{aligned}
x' &= x(r_{11} + r_{12}w_z - r_{13}w_y) + y(-r_{11}w_z + r_{12} + r_{13}w_x) \\
&\quad + z(r_{11}w_y - r_{12}w_x + r_{13}) + r_{11}t_x + r_{12}t_y + r_{13}t_z + \tau_x \\
y' &= x(r_{21} + r_{22}w_z - r_{23}w_y) + y(-r_{21}w_z + r_{22} + r_{23}w_x) \\
&\quad + z(r_{21}w_y - r_{22}w_x + r_{23}) + r_{21}t_x + r_{22}t_y + r_{23}t_z + \tau_y \\
z' &= x(r_{31} + r_{32}w_z - r_{33}w_y) + y(-r_{31}w_z + r_{32} + r_{33}w_x) \\
&\quad + z(r_{31}w_y - r_{32}w_x + r_{33}) + r_{31}t_x + r_{32}t_y + r_{33}t_z + \tau_z
\end{aligned}
$$

and where x', y' and z' are the coordinates of the 3D object after the motion described in equation (5.17) in the ith camera's view, and f_i is the focal length of the ith camera. Just as in the single camera case, we wish to compute the entries of $\boldsymbol{F_{i\mu}}\Big|_{\Delta\boldsymbol{\mu}=0}$ (which will be referred to as $\boldsymbol{F_{i\mu}}$ from this point on):

$$
\boldsymbol{F_{i\mu}} = \begin{bmatrix} u_1 & u_2 & u_3 & u_4 & u_5 & u_6 \\ v_1 & v_2 & v_3 & v_4 & v_5 & v_6 \end{bmatrix} \frac{f_i}{(z'_\mu)^2} \tag{5.20}
$$

where $(z'_\mu)^2$ represents the derivative of z' w.r.t. $\boldsymbol{\mu}$ evaluated at $\boldsymbol{\mu} = 0$ and u_i and v_i are the derivatives of x' and y' w.r.t. the parameters of $\boldsymbol{\mu}$. The form of $\boldsymbol{F_{i\mu}}$ comes from the result of these derivatives. For example, the derivative of the point $\frac{x'}{z'}$ w.r.t. w_x is

$$
\frac{d}{dw_x}\left(\frac{x'}{z'}\right) = \frac{u_1}{(z'_{\boldsymbol{\mu}})^2} \tag{5.21}
$$

where

$$u_1 = \frac{d}{dw_x}(x') * z' - x' * \frac{d}{dw_x}(z'). \tag{5.22}$$

The remaining eleven derivatives are similarly computed. Therefore, to compute the entries of $\boldsymbol{F_{i\mu}}$, one needs to compute the derivatives of x', y' and z' w.r.t. each parameter of M and then evaluate each expression at $\boldsymbol{\mu} = 0$. As an example, the derivatives of x' w.r.t. w_x and t_x are given.

$$\begin{aligned}\frac{dx'}{dw_x} =& x * (r'_{13} * r_{21} - r'_{12} * r_{31}) + \\ & y * (r'_{13} * r_{22} - r'_{12} * r_{32}) + \\ & z * (r'_{13} * r_{23} - r'_{12} * r_{33}) - \\ & r'_{12} * \tau_z + r'_{13} * \tau_y \end{aligned} \tag{5.23}$$

$$\frac{dx'}{dt_x} = r'_{12}, \tag{5.24}$$

where x, y, and z are the original 3D coordinates of the rigid object. The remaining derivation to obtain the entries of $\boldsymbol{F_{i\mu}}$ is left to the reader in lieu of space.

In the final step of full-motion recovery from multiple cameras, a single $\boldsymbol{\Delta\mu}$ is computed from all camera views. To do this, the spatial image gradients ($\boldsymbol{I}_p$), the temporal image gradients ($\boldsymbol{I}_p$), and the partial derivatives of the 2D parametric motion ($\boldsymbol{F_\mu}$ and $\boldsymbol{F_{i\mu}}$) must be combined. This is done in the following manner. First, we calculate $\boldsymbol{G'}$, a multiple camera version of the $\boldsymbol{G}$ shown in equation (5.6), as

$$\boldsymbol{G'} = [\boldsymbol{G_1 G_2 ... G_m}]^T \tag{5.25}$$

where $\boldsymbol{G_i}$ refers to the calculation for $\boldsymbol{G}$ from equation (5.6) for the ith camera in a multiple camera system with m cameras. For instance, $\boldsymbol{G_2}$ will be calculated for the second camera by $\boldsymbol{G_2} = \boldsymbol{I_{2_p} F_{2_\mu}}$. Then, $\boldsymbol{G'}$ will represent the spatial image gradients and partial derivatives of the 2D parametric motion for all of the cameras in the system. Similarly, the temporal image gradients from all of the cameras in the system for the current image are concatenated to form $\boldsymbol{I'_t}$. Now, to compute the global $\boldsymbol{\Delta\mu}$, $\boldsymbol{G'}$ from equation (5.25) is substituted in for $\boldsymbol{G}$ and $\boldsymbol{I'_t}$ is substituted for $\boldsymbol{I_t}$ in equation (5.5).

The $\boldsymbol{\Delta\mu}$ computed above represents the motion of the 3D object in the first camera's view, since we have considered that view the same as the world coordinates. For the ith camera, an additional step is needed. First, $\boldsymbol{M}$ is formed from $\boldsymbol{\Delta\mu}$. Then $\boldsymbol{M}$ is used in equation (5.17) to recover $\boldsymbol{X}_i(t+1)$ after the motion has taken place. This, of course, may be repeated for any number of cameras in a multiple camera system.

5.2.3 3D Cylinder Head Model

The above methods for recovering 3D motion in single and multiple cameras may be used with any arbitrary rigid object. In our experiments, a 3D cylinder was chosen as the model. One reason for the choice of this shape is for its ease in initialization and close approximation to the head. Also, by unwrapping the cylinder texture of the face, a suitable image for face

recognition is recovered, which is discussed in further detail in section 5.3.3.

The cylinder is initialized manually by adjusting the size and position of the cylinder on the face in the first frame of the video sequence and then adjusting the pose to match the pose of the face. This is done in our experiments by using an estimated $\boldsymbol{C}_i$ between each camera i and the world coordinate system (first camera). The estimation is done using OpenCV and a checker board pattern similar to that found in Figure 5.17(a). However, since the cylinder model must be initialized at the beginning of the face tracking task, the initialization from multiple cameras may be used to estimate the rotation and translation matrix $\boldsymbol{C}_i$ that is used in the multiple camera full-motion recovery model.

Cylinder motion in two camera views is displayed in Figures 5.14 and 5.15. Figures 5.14(a) and 5.15(a) display the cylinder at time t from the two camera views. Figures 5.14(b) and 5.15(b) display the same cylinder at time $t + 1$ (1 second) after motion $\boldsymbol{M}$ has been applied to it.

5.2.4 Occlusion

In this work, we are concerned with handling two main types of occlusion. The first type, self-occlusion, is very common in face tracking tasks. If the pose of the face is nonfrontal, particularly when the pose is larger than around +/- 30 degrees in yaw and/or tilt from the frontal pose, pixels of the face that have been used for tracking will not be present and therefore may affect the tracking result greatly. The second type is full face occlusion, meaning

that the entire face is occluded in one or more cameras.

Several approaches have been tried in handling occlusion in face tracking. In [91], an occlusion pixel is detected by comparing the motion residual error of each pixel with all pixels used in the tracking. If the motion residual error is larger than a threshold, the pixel is classified as an outlier or occlusion. In some works such as in [54], occlusion is detected simply by calculating the number of skin color pixels. If there is very small portion of skin pixels in the target, occlusion is detected.

5.2.4.1 Self-Occlusion

We handle the case of self-occlusion in our tracking model by weighting a mask that is used in the calculation of the temporal image gradient for each camera's image. For a frontal face pose, the mask is centered on the face and the most weight is given to the center pixels and the least to pixels furthest away from the center using Gaussian weights. If the pose of the face is nonfrontal, the mask is centered on the part of the face that is most visible to the camera's view. Therefore, pixels that are most visible are weighted the most in the face tracking calculation and the camera with the most frontal view of the face will be weighted the most in our multi-camera tracking method. The weighted mask is updated for each frame based on pose estimation.

5.2.4.2 Full Face Occlusions

While the above method is beneficial for tracking the face, even across face poses that are far from frontal, it is not sufficient for handling partial or full occlusion of the face. Our method for handling face occlusion relies on multi-camera tracking. Occlusions of the face are handled by first detecting the occlusion in a particular camera and then removing that camera from the multi-camera motion calculation.

To determine if a camera is being occluded, we use a comparison of image histograms approach. First, the unwrapped cylinder texture image is taken from the template (usually the first image in the sequence) and the current frame. An example template image is found in Figure 5.4(a), while a non-occluded current frame and an occluded current frame are found in Figures 5.4(b) and 5.4(c), respectively.

The histograms of the images are formed from the H channel of the HSV color map of the image and are shown in Figures 5.5 and 5.6.

To measure the similarity of the two histograms (one from the template image and the other from the current frame), we employ the Bhattacharyya coefficient [9] , which is a measure of the amount of overlap between two statistical samples and is commonly used to measure the similarity of histograms. Figures 5.7, 5.8 and 5.9 display the Bhattacharyya coefficients for cameras 1, 2 and 3, respectively. The dashed line in each of the plots is the threshold value which was determined as two standard deviations from the mean when

no occlusion are present. This threshold, once found, was used throughout the experiments. From the plots, it is clear that camera 3 has detected an occlusion.

(a) (b) (c)

Figure 5.4: Example template, non-occluded and occluded frames

5.3 Experimental Results

We will discuss three experiments that display the efficacy of our multi-camera full-motion recovery model. In the application area of pose estimation, we show the advantage of using the multi-camera model over the single camera model using a two-camera video sequence that contains pose ground truth of the face. We also test our occlusion robust face tracking on three-camera video sequences that contain both self occlusion and camera occlusion. Finally, the face recognition experiment also shows the advantage of our method in a real application.

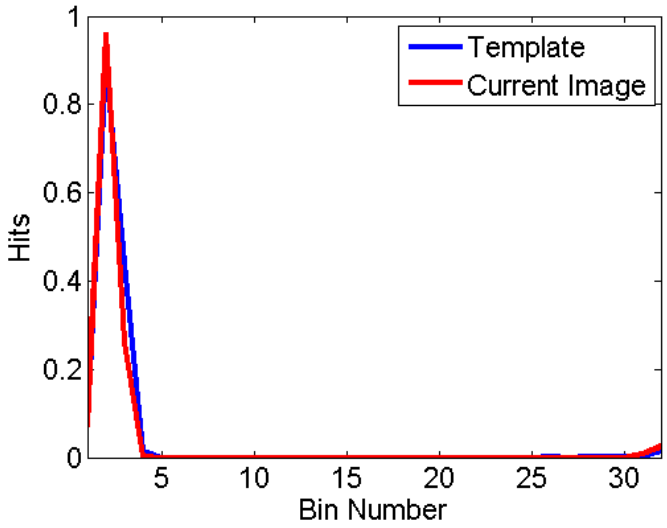


Figure 5.5: Histogram of template and current frame

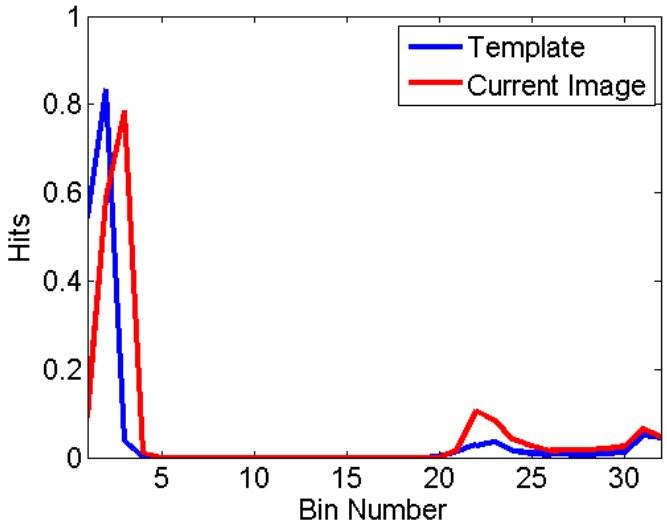


Figure 5.6: Histogram of template and partially occluded frame

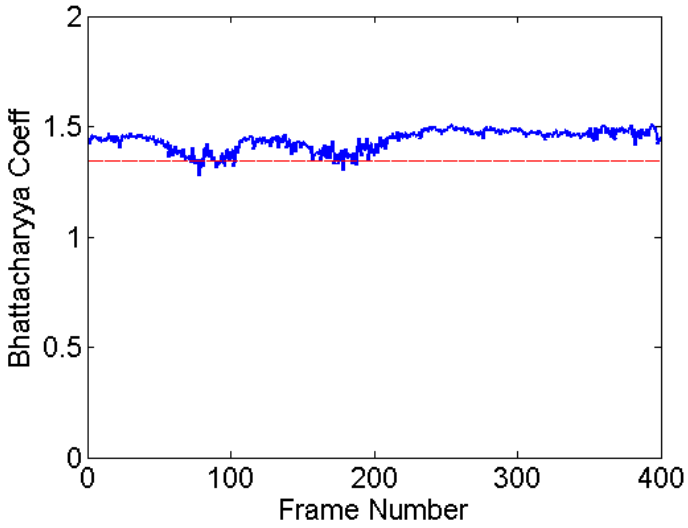


Figure 5.7: Bhattacharyya coeff for camera 1

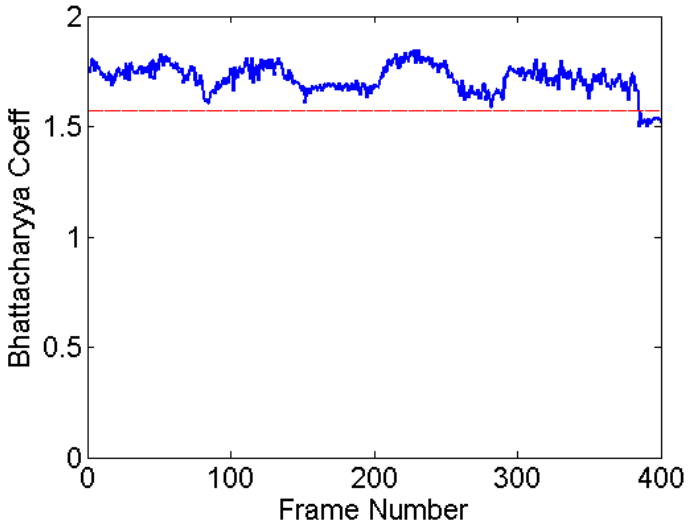


Figure 5.8: Bhattacharyya coeff for camera 2

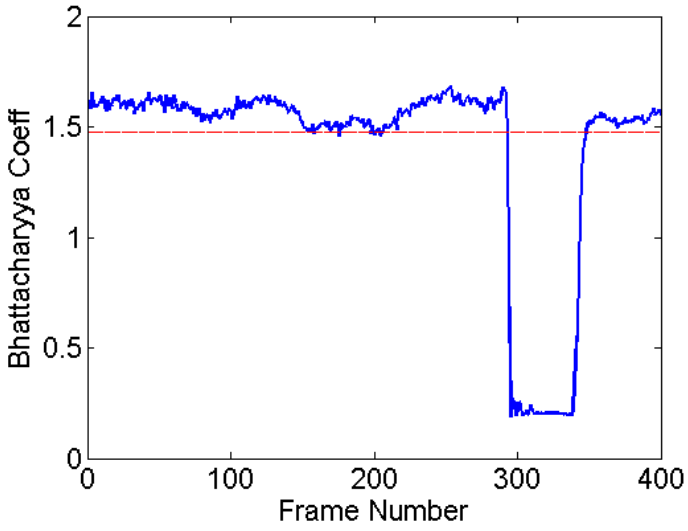


Figure 5.9: Bhattacharyya coeff for camera 3

5.3.1 Pose Estimation from Unoccluded Cameras

The advantage of our multiple camera approach to full recovery of motion of the face is shown when tracking in a realistic setting. For this experiment, a video sequence of an individual was obtained from two cameras. To generate ground truth, a checkerboard pattern was placed on the head of the subject and a camera calibration package was utilized to obtain the rotation vector of the checkerboard in each frame [10]. The results for the yaw and pitch from the face tracking experiment are shown. Figures 5.10 and 5.11 display the results of the yaw for the left and right cameras, while Figures 5.12 and 5.13 display the results of the tilt for the left and right cameras, respectively. Figures 5.14 and 5.15 display images from the tracking results

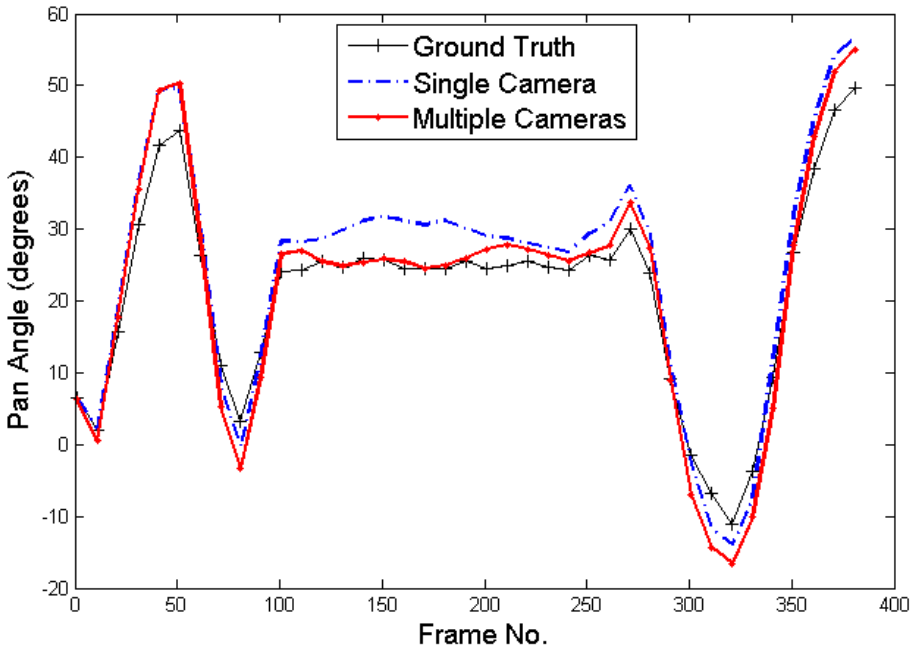


Figure 5.10: Yaw estimation of face tracking from left camera

from the single (top) and multiple (bottom) camera models from the right and left cameras, respectively. In addition to these results, video sequences of two-camera and three-camera tracking that visually display the advantages of our multiple camera method over single camera tracking may be found on our web page [33].

Table 5.1 displays the mean squared error (MSE) and mean absolute error (MAE) of the pan and tilt angles comparison between the single camera and multiple camera models of the left and right cameras.

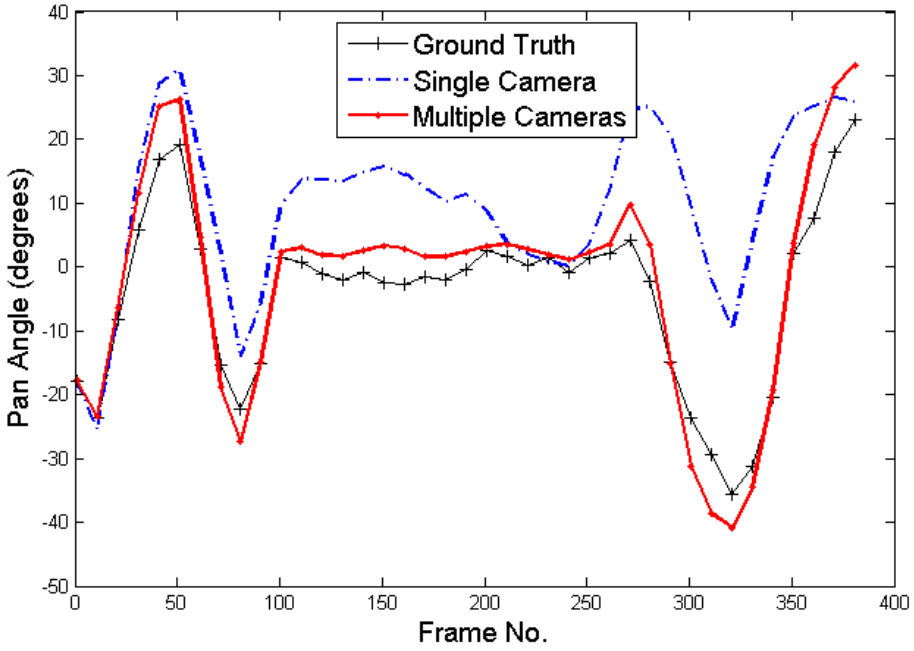


Figure 5.11: Yaw estimation of face tracking from right camera

Table 5.1: Comparison of mean squared error (degrees squared) and mean absolute error (degrees) of pan and tilt between single and multiple camera models

Model	Rotation	Camera	MSE	MAE
single	pan	left	21.8	4.2
	pan	right	300.5	13.9
	tilt	left	18.8	3.5
	tilt	right	224.4	11.2
multi	pan	left	14.0	2.9
	pan	right	23.2	3.8
	tilt	left	13.9	3.2
	tilt	right	17.7	3.3

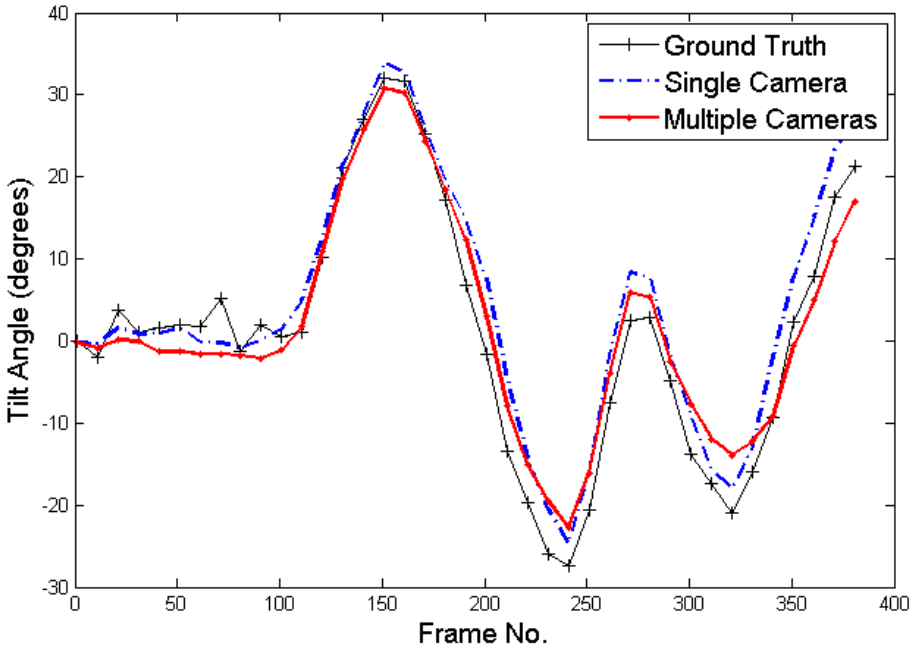


Figure 5.12: Pitch estimation of face tracking from left camera

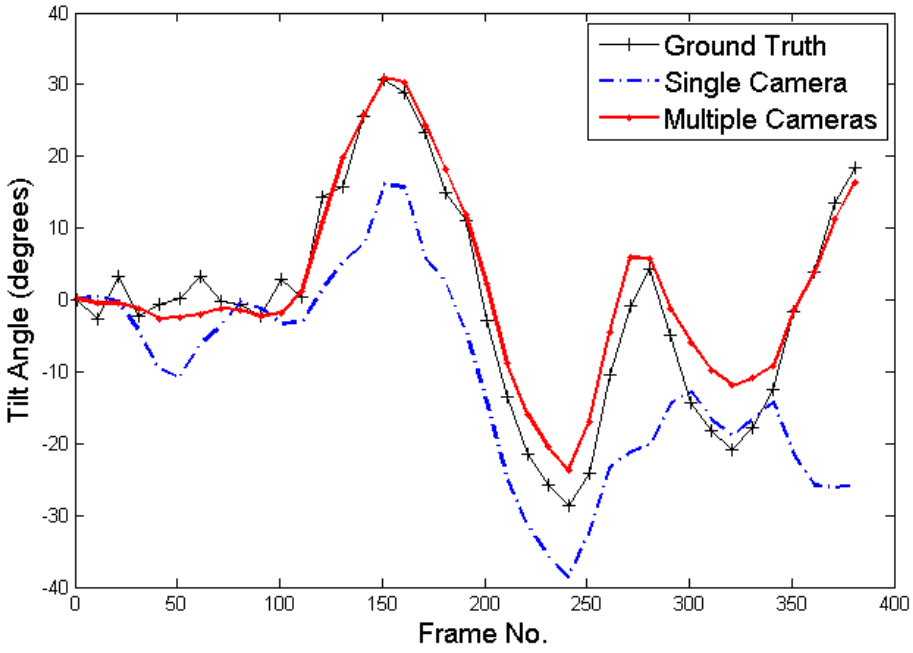


Figure 5.13: Pitch estimation of face tracking from right camera

5.3.2 Pose Estimation from Occluded Cameras

In each of the experiments, the cylinder is initialized manually by adjusting the size and position of the cylinder on the face in the first frame of the video sequence and then adjusting the pose to match the pose of the face. The rotation and translation parameters between the cameras are then used to refine the initialization so that the cylinder is initialized in an acceptable position for all cameras.

For these experiments, two video sequences of an individual was obtained from three cameras. The first video sequence was used to estimate the threshold for detecting occlusion and then towards the end of the sequence an occlusion in camera 3 is presented. In the second video sequence, an occlusion is presented in camera 2. In our multi-camera system, camera 1 is the center-most camera, camera 2 is the right-most camera, and camera 3 is the left-most camera. To generate ground truth, a checkerboard pattern was placed on the head of the subject and OpenCV [11] was used to obtain the rotation vector of the checkerboard in each frame.

Figures 5.18, 5.19 and 5.20 display the results of the yaw for cameras 1, 2 and 3, while Figures 5.21, 5.22 and 5.23 display the results of the tilt for cameras 1, 2 and 3, respectively. In each of the plots, a camera that has lost track of the face is denoted by a constant value for the pose after the track is lost. An instance of this is apparent in Figure 5.23 where the single camera has lost track of the face in frame number 300. In addition to these results, we will submit the video sequences of three-camera tracking to visually display

Table 5.2: RMS error between estimated pose and ground truth for yaw (degrees)

Camera	Method	Sequence 1	Sequence 2
1	Single	5.28	3.49
	Multi	6.52	24.93
	MultiOcc	4.27	4.74
2	Single	13.46	6.67
	Multi	6.44	7.78
	MultiOcc	3.89	1.10
3	Single	4.96	6.44
	Multi	4.05	24.98
	MultiOcc	3.45	4.66

the advantages of our multiple camera method over single camera tracking.

Tables 5.2 and 5.3 display the root mean squared error (RMS) of the pan and tilt angles comparison between the single camera (Single), multiple camera (Multi) and occlusion robust multiple camera (MultiOcc) tracking of the multi-camera system. The RMS value was computed for each camera and method using the least number of frames that were successfully tracked for all three cameras. For example, in sequence 2, camera 2 is unable to track the face past frame 71. Therefore, only frames 1 through 71 were used for all three cameras in the RMS calculation. Therefore, the RMS values unfairly penalize the occlusion robust multi-camera face tracking method when the single camera tracking has failed.

Table 5.3: RMS error between estimated pose and ground truth for tilt (degrees)

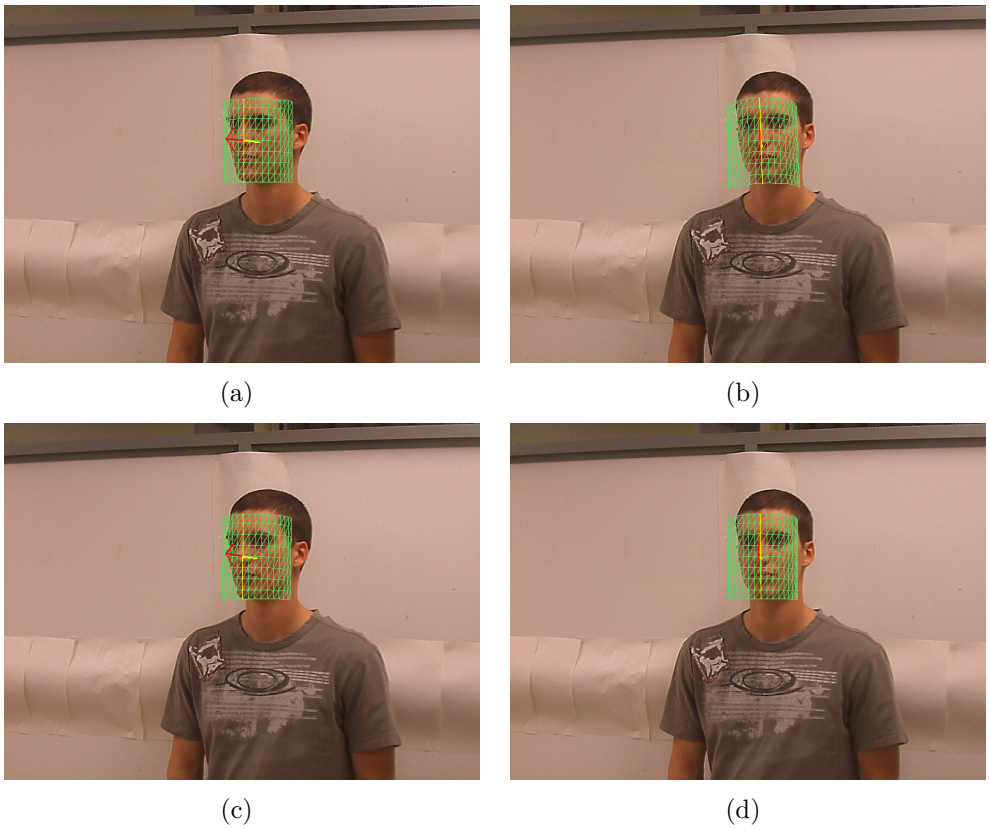
Camera	Method	Sequence 1	Sequence 2
1	Single	8.22	3.72
	Multi	8.57	6.77
	MultiOcc	8.36	3.42
2	Single	16.43	3.53
	Multi	7.95	4.81
	MultiOcc	8.20	3.96
3	Single	18.62	3.13
	Multi	7.95	7.36
	MultiOcc	7.99	3.25

5.3.3 Face Recognition

A real world application of the full-motion recovery of the face from multiple cameras is in face recognition. Using the methodology similar to that in [37], we show that face recognition may be improved using the multiple camera model when compared to the single camera model. This is intuitive, since a better face tracking result should produce images more suitable for recognition in a traditional frontal face still image face recognition system.

Face recognition in our experiments is performed in the following manner for our two camera video sequences. First, the face is tracked using the single camera model and the multiple camera model in both video sequences. Figure 5.14 and 5.15 display example images from the tracking result used from one subject of our database. Each frame in the face tracking sequence produces a cylinder texture map of the face which provides as much of a frontal

view of the face as possible with the 3D cylinder model. This resulting cylinder texture map is then cropped and used for recognition. An example of such an image is shown in Figure 5.16. Eigenfaces [79] is used for it's simplicity as a benchmark algorithm to test our methodology, but any still face recognition method may be used.

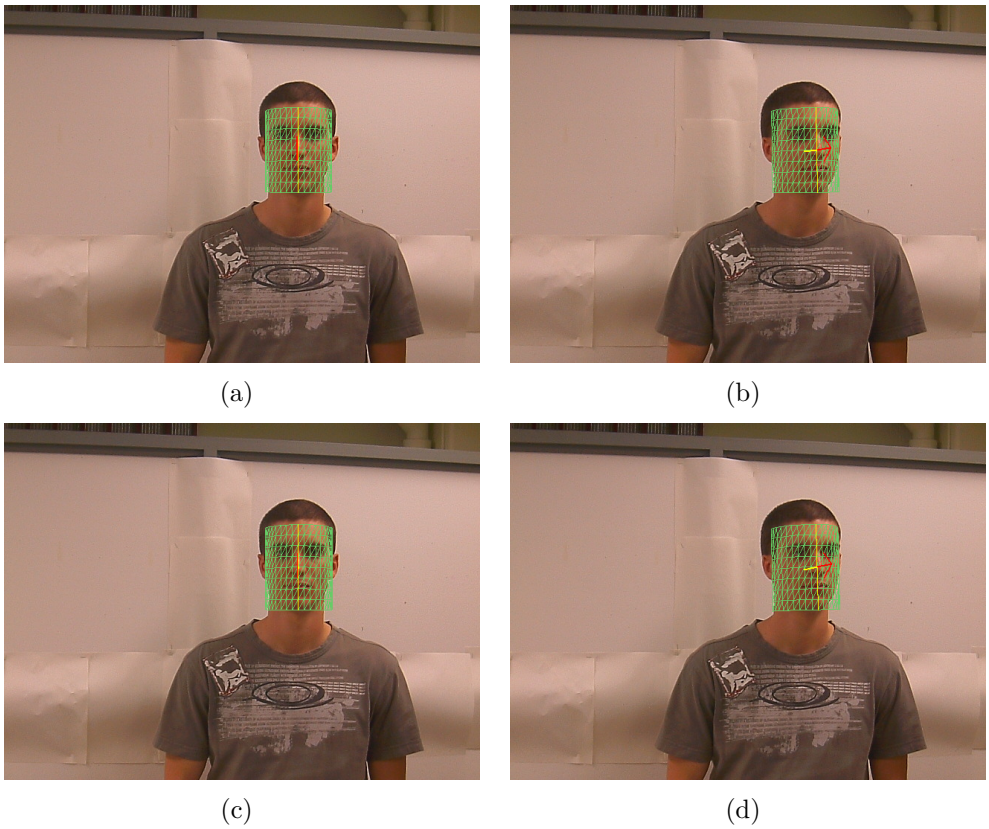


(a) (b)

(c) (d)

Figure 5.14: Cylinder tracking result for single (a & b) & multiple (c & d) camera motion from second (right) camera

Our data consists of 100 frames per subject from a video sequence of 20 seconds over 18 subjects from two cameras. Only one frontal image per



(a) (b)

(c) (d)

Figure 5.15: Cylinder tracking result for single (a & b) & multiple (c & d) camera motion from first (left) camera

subject was used for training and the remaining data was used for testing. The results from our face recognition experiment are in Table 5.4.

The columns labeled "none" and "mindist" refer to whether or not the results from the two cameras were combined and how, as described in [37]. "None" refers to using all of the images for face recognition regardless of their source while "mindist" refers to using a minimum distance nearest neighbors classifier to decide the face recognition result between two cameras at the same

Figure 5.16: Example of face from cylinder texture map

Table 5.4: Face Recognition Results from Single and Multiple Camera Models

model	none	mindist
single	68.5	82.8
multi	71.2	84.7

time frame. In either case, it is apparent the multiple camera model produces images that are more suitable for still face recognition.

5.4 Discussion

The motion of a face as viewed from multiple cameras intuitively gives more information than that of a single camera. Using this information explicitly, we have formed a model for full-motion recovery of the face from multiple cameras. This multiple camera model outperforms the single camera model in pose estimation and face recognition.

In regards to pose estimation from the unoccluded cameras, the multiple camera model provided results closer to ground truth than the single

camera model. The results reported in Table 5.1 clearly display the overall increase in robustness of the tracking result when using the multiple camera model over the single camera model. It is worth mentioning that the right camera produces more error in pose estimation than the left camera (as in Figures 5.11 and 5.13) because the initialization of the cylinder is performed on an image of a nonfrontal face, as seen in Figure 5.14(a).

In regards to pose estimation from occluded cameras, it is clear from the results of the pose estimation plots and the RMS error of each method with ground truth that the occlusion robust multi-camera face tracking is far superior to the single camera method. In both sequences there is at least one camera that loses track of the face with single camera tracking. In sequence 2, even the multi-camera face tracking method has trouble with the occlusion. It is important to note that when the multi-camera face tracking approach fails, it loses track of the face in all three cameras since they are all connected by a single motion calculation. However, by explicitly handling occlusion, the cameras work together to maintain a robust tracking of the face in the proposed method. Although our pose estimation experiments are shown with two and three cameras, the methodology may easily be extended to any number of cameras.

It is clear from the face recognition experiment that the multiple camera model produces images that are more suitable for still face recognition and thus improve the accuracy of the recognition result. It is worth noting that in this experiment, the video sequence was chosen so that the single camera model did

not lose track of the face to provide a fair comparison of the quality of image that is produced for face recognition between the two methods. Obviously, if the single camera model loses track of the face, the recognition results would suffer greatly. This is seen most easily in Figure 5.17, where the masked texture produced from the single camera tracking (Figure 5.17(e)) is not suitable for face recognition (and is incorrectly labeled by our face recognition system) while the texture from the multiple camera tracking of the same frame (Figure 5.17(f)) is recognized correctly by our face recognition system.

5.5 Summary

A novel approach to robust object tracking by full-motion recovery from multiple cameras is presented. This approach builds on the single camera model by explicitly including the motion from multiple cameras into a joint motion calculation. The proposed occlusion robust multi-camera face tracking method has been shown to be robust to self-occlusion and full face occlusion and significantly outperforms the single camera tracking from each of the cameras. The multiple camera full-motion recovery model improves face tracking over a single camera as is shown in our experiments on pose estimation and face recognition. Future work on this topic includes improving the face tracking by including more cameras in our system, applying the motion model to other 3D shapes and implementing automatic initialization.

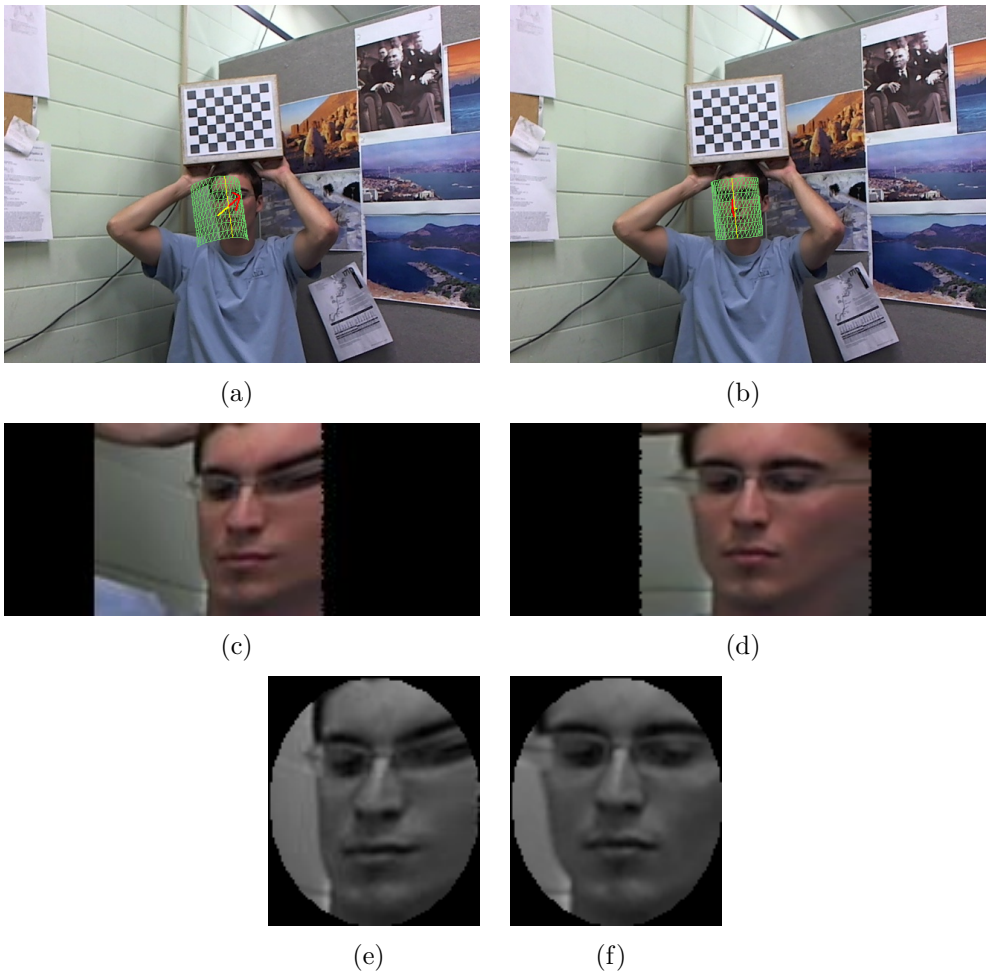


Figure 5.17: Cylinder tracking result for single (a) & multiple (b) & camera motion, the extracted textures (c) & (d), and the masked images used for face recognition (e) & (f), respectively

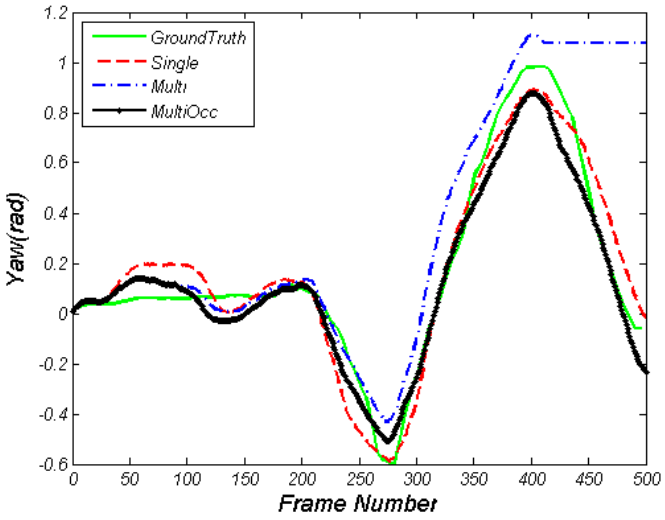


Figure 5.18: Yaw estimation of tracking sequence 1 from camera 1

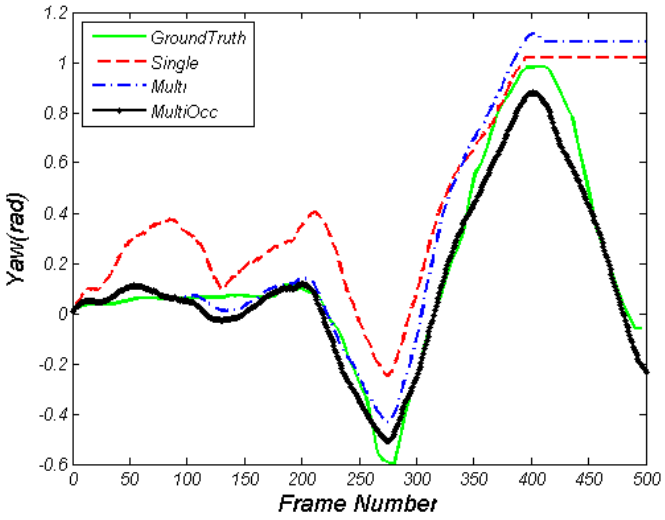


Figure 5.19: Yaw estimation of tracking sequence 1 from camera 2

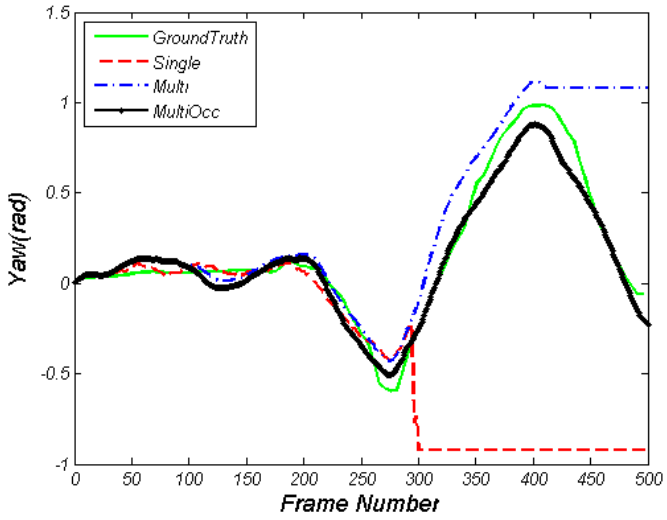


Figure 5.20: Yaw estimation of tracking sequence 1 from camera 3

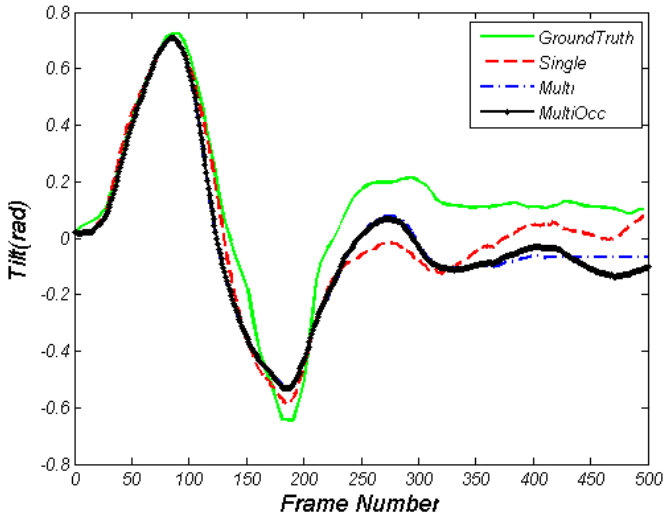


Figure 5.21: Tilt estimation of tracking sequence 1 from camera 1

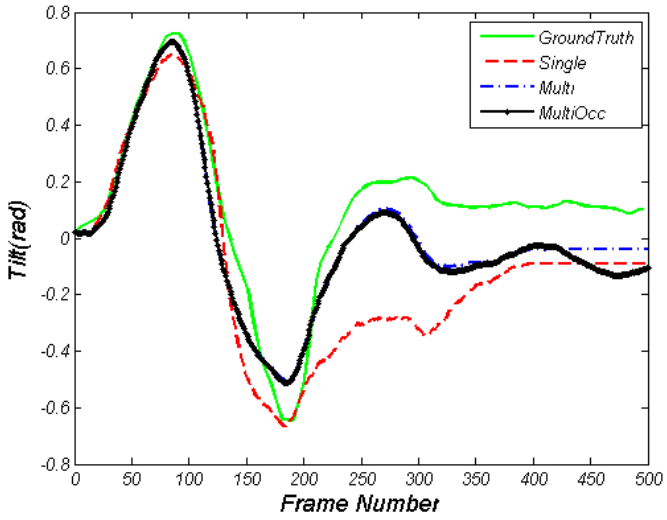


Figure 5.22: Tilt estimation of tracking sequence 1 from camera 2

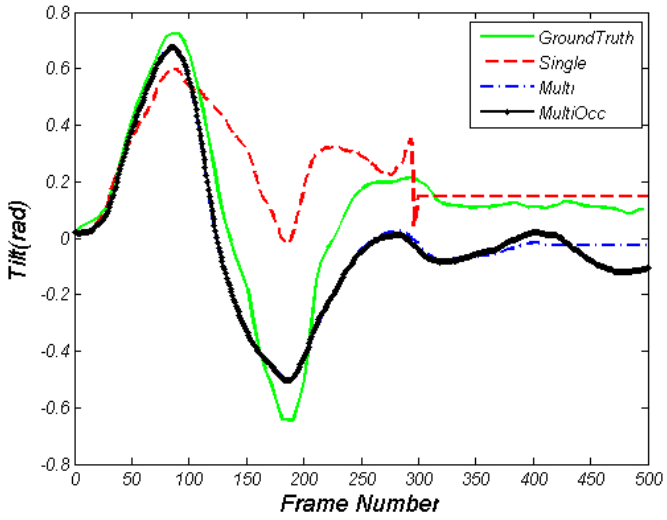


Figure 5.23: Tilt estimation of tracking sequence 1 from camera 3

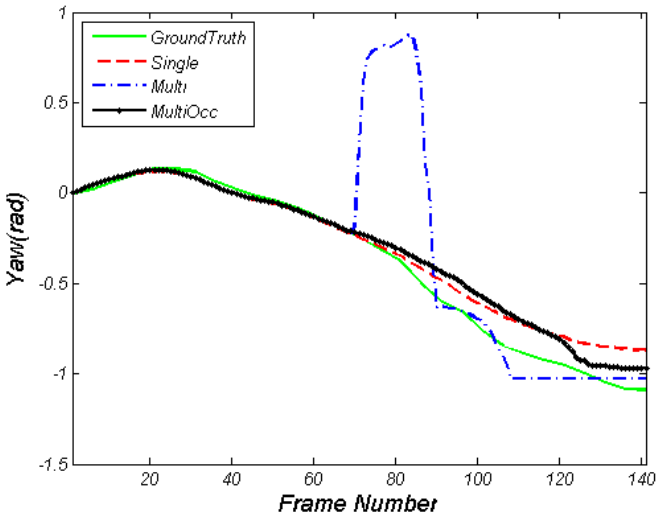


Figure 5.24: Yaw estimation of tracking sequence 2 from camera 1

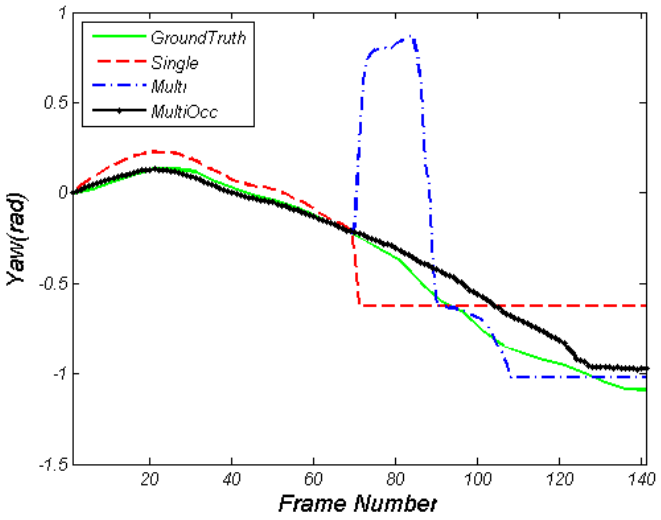


Figure 5.25: Yaw estimation of tracking sequence 2 from camera 2

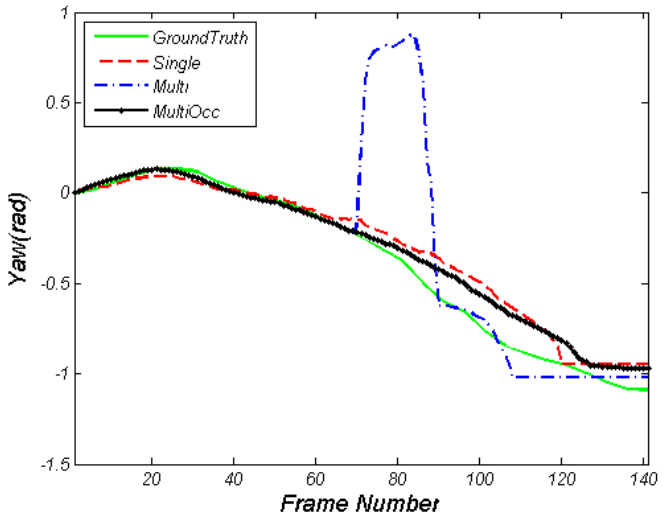


Figure 5.26: Yaw estimation of tracking sequence 2 from camera 3

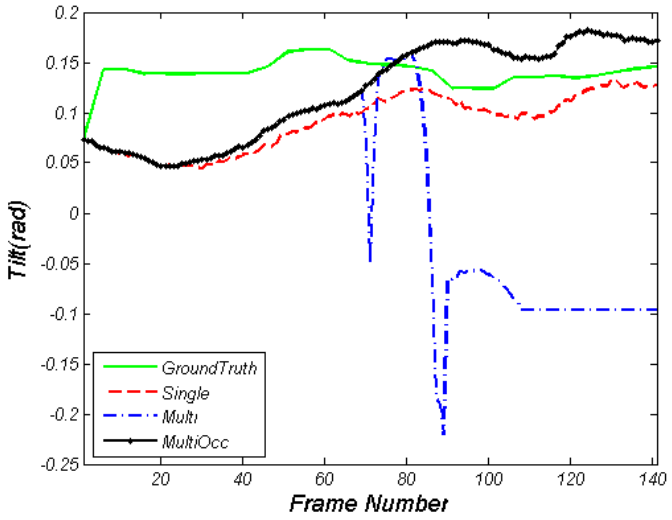


Figure 5.27: Tilt estimation of tracking sequence 2 from camera 1

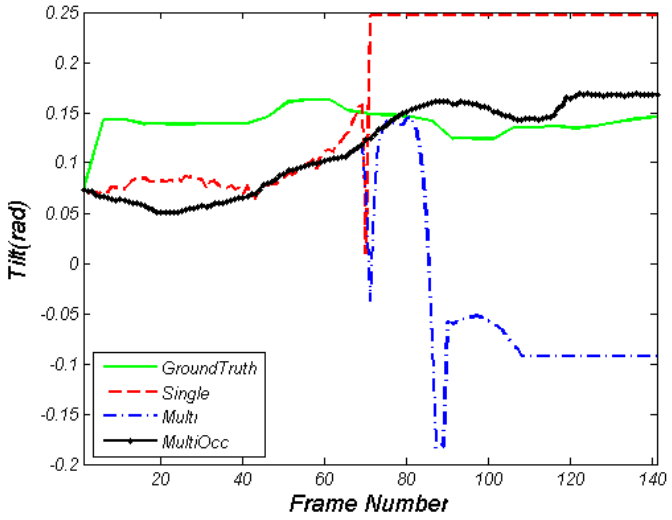


Figure 5.28: Tilt estimation of tracking sequence 2 from camera 2

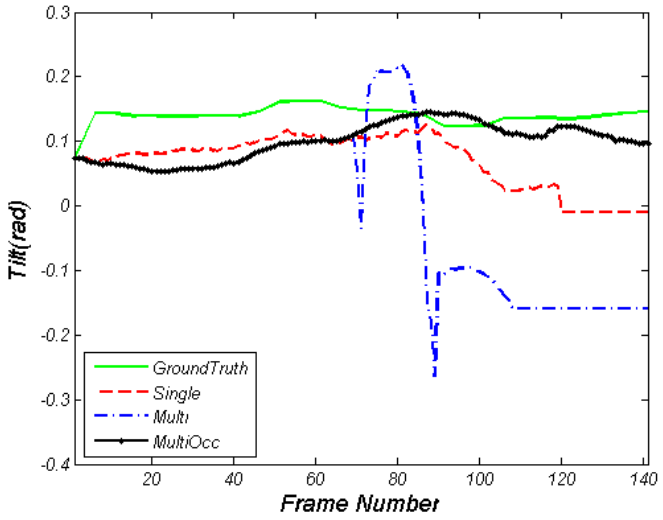


Figure 5.29: Tilt estimation of tracking sequence 2 from camera 3

Chapter 6

Still Face Recognition with the Average-Half-Face

The work presented up to this point of the dissertation has focused on the problem of face recognition from video. As noted in the introduction, advances that are made to the area of still face recognition may also assist in solving face recognition from video. Another theme of this dissertation is fusing information from multiple sources to improve the final result, whether that be tracking, pose estimation or face recognition. In this spirit, we introduce our work on the average-half-face which enhances the accuracy of still face recognition by fusing the information from both halves of the face.

It is well known that the human face is inherently symmetric about a bilateral symmetry axis. This symmetry has been utilized in many applications pertaining to face detection and recognition. The average-half-face is one such utilization of symmetry and has been shown to be successful in the face recognition task. To construct the average-half-face, the right and left half-face are averaged together, assuming the bilateral symmetry axis has been found. We present the results applying the average-half-face to two- and three-dimensional (2D and 3D) face databases by utilizing several popular face

recognition algorithms. Our results show that the average-half-face produces a clear increase in accuracy in most cases. This finding may have implications on storage and computation time for face recognition systems. Further, since the success of using the average-half-face in face recognition may depend heavily on the computation of the bilateral symmetry axis of each face, we present an error analysis of choosing this bilateral symmetry axis and find that the average-half-face is more robust to this choice than the original full face. Finally, we present a study to measure and analyze the symmetry of the face and its effect on face recognition accuracy.

6.1 Symmetry of the Face

Several authors have noted the role of symmetry in nature [23, 25] and particularly in human face attractiveness [41, 68, 78]. It is noted and often observed that the "face is roughly symmetrical" [74]. Additionally, research into using the symmetry of the face to assist with face detection and face recognition has been previously studied. Chen *et al.* [17] developed a method to automatically compute the bilateral symmetry axis (or plane for 3D data), which is particularly useful in this research. The symmetry of the face is used by Zhao and Chellappa [94] to detect and remove illumination effects in faces to improve recognition with what they term Symmetric Shape-from-Shaping. The use of symmetry has also proven useful to extract the facial profile for face recognition, such as in [65, 90]. In [67], Ramanathan *et al.* perform similarity measures between images of the same individual to study the affects of age,

disguise, illumination and pose on the face using the notion of 'Half-faces'. Our work is different in that we seek to utilize the symmetry of the face in a holistic manner for face recognition and to study the symmetry of the face between subjects and the overall impact of symmetry on face recognition.

6.2 Average-Half-Face

The original inspiration for the average-half-face [34–36] is the symmetry preserving singular value decomposition (SPSVD) [72]. The SPSVD is used in place of the singular value decomposition (SVD) when the data contains inherent symmetry. Therefore, the SPSVD is used to reduce the dimensionality of data while preserving the inherent symmetry. When applying the concept of the SPSVD to a 2D image, such as that of a face, there are two steps to accomplish this. First, once the bilateral symmetry axis of the face has been calculated, the image is centered about the bilateral symmetry axis (usually the nose of the (properly oriented) face). This step will preserve the symmetry and ensure that the two spatial halves of the data are near similar (mirrored) images. Second, the face image is divided into two halves and they are averaged together by first reversing the columns of one of the halves. In our experiments, we do not desire a dimensionality reduction on the face image or the average-half-face, so the step of performing the (SVD) on the image is skipped.

As an example of the process of forming the average-half-face, Figure 6.1 displays the full face image, the left and right faces (after centering based

on the bilateral symmetry axis), and the average-half-face of an image from the Yale Face database [1].

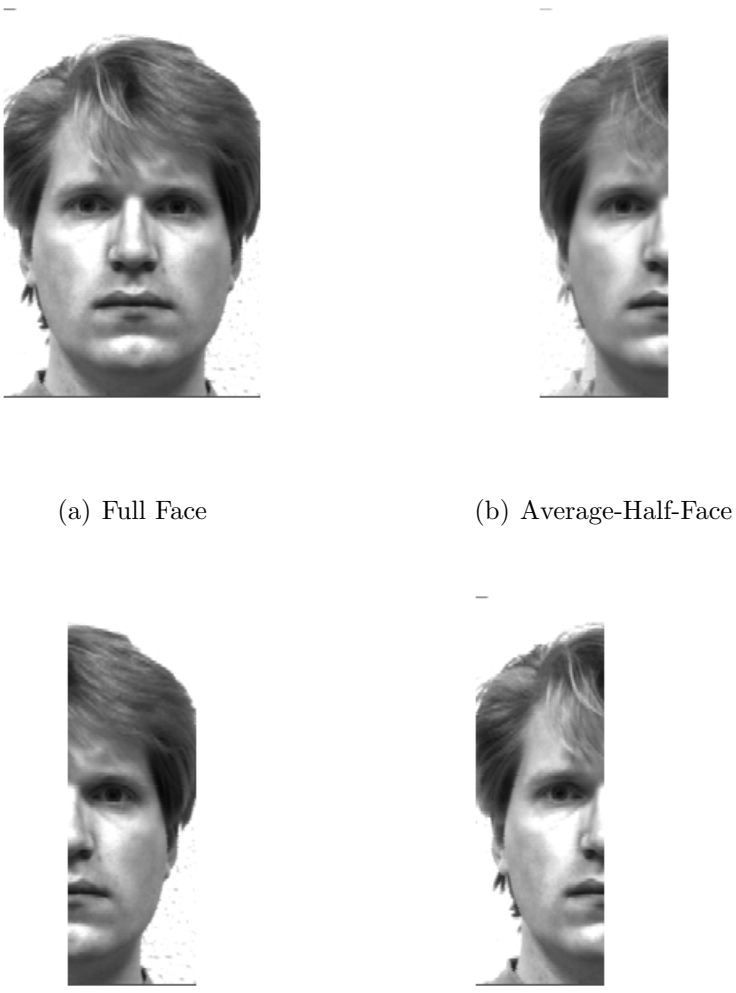
Since these steps are independent of the face recognition algorithm, the average-half-face can be seen as a preprocessing step. This allows for a direct comparison of the use of the average-half-face with the original full face image in face recognition algorithms.

6.3 Face Recognition Algorithms

Each of the six face recognition methods that will be used in our experiments have been previously introduced in Section 2. Please refer to that section for more information regarding these methods. We will be utilizing the following methods for our experiments:

1. Eigenfaces (PCA)
2. Multilinear PCA (MPCA)
3. Multilinear PCA + LDA (MPCA-LDA)
4. Fisherfaces (LDA)
5. Independent Components Analysis (ICA)
6. PCA + Support Vector Machines (SVM)

The first five methods classify the test samples using nearest neighbors with Euclidean distance while the sixth method classifies using SVM. For



(a) Full Face

(b) Average-Half-Face

(c) Left Half-Face

(d) Right Half-Face

Figure 6.1: (a) 2D full face image; (b) its average-half-face; (c) its left half-face; and (d) its right half-face.

implementation of the SVM, we utilized the powerful LIBSVM library [14] that uses the "one vs. one" approach to multi-class problems. We used default parameters for our SVM, including a radial basis function (rbf) kernel.

6.4 Databases

Three face databases were used for our experiments; A: The Yale Face database, B: The AR Face database, and C: 3D face database.

The Yale Face database (A) [1, 8] consists of a total of 165 gray scale, frontal, 2D face images. There are a total of 15 subjects with 11 images per subject representing changes in illumination and facial expressions. For each of the algorithms, we maintained a consistent use of the database by forming the training data from the first 8 images per subject and using the remaining 3 images per subject for testing.

We used images from 109 subjects (66 men and 43 women), each with 26 configurations from the AR Face database (B) [61]. The different configurations consist of expression changes (such as neutral, smile, anger, and scream), lighting changes, and occlusions. Two different sessions, each with 13 different configurations, were taken to form the database. We used the first 21 configurations per subject for training and the remaining images for testing.

We have additionally utilized a 3D face range image database (C) called the Texas 3D Face Recognition Database (Texas 3DFRD) [29–31] acquired using an MU-2 stereo imaging system manufactured by 3Q Technologies Ltd.

(Atlanta, GA) by the former company Advanced Digital Imaging Research, LLC, Friendswood, TX. The database consists of a total of 1126 images of 104 subjects. There are anywhere from 1 to 55 images per subject. We trained the algorithms using a combination of 360 images from 12 subjects and a single neutral expression from 104 different subjects. The test database consisted of the remaining 662 images from all 104 subjects.

6.5 Experiments

In total, 18 experiments were performed using the six face recognition methods and three face databases. In addition to these experiments, an error analysis was done on the choice of the bilateral symmetry axis of the face.

6.5.1 Varying Algorithms and Databases

The parameters for each algorithm were kept constant between experiments to maintain a fair comparison of each algorithm's performance on the average-half-face and the full face. Also, the images were centered for both the average-half-face and full face recognition results. Table 6.1 summarizes the results of our experiments. Each of the numbers in the table represents the rank-1 accuracy rate for recognition, meaning that we report only the accuracy of the closest match of the test data to a corresponding training sample. It is crucial to recall when studying the table that the purpose of these experiments is to compare the full face to the average-half-face for recognition, not to compare the accuracy of the algorithms themselves.

Table 6.1: Rank-1 accuracy results using the full face (Full) and the average-half-face (AHF).

Database	A		B		C	
	Yale		AR		3D	
Algorithms	Full	AHF	Full	AHF	Full	AHF
PCA	77.8	86.7	49.4	52.3	72.8	80.4
MPCA	80.0	93.3	59.4	57.6	81.0	81.3
MPCA-LDA	66.7	68.9	91.9	88.1	91.8	93.8
LDA	91.1	97.8	54.1	78.0	79.8	82.6
ICA	93.3	100	65.3	60.0	76.9	84.3
SVM	91.1	91.1	44.8	36.1	50.8	51.4

Figures 6.2, 6.3, and 6.4 display these results more clearly for each of the three databases involved. From the results of Figures 6.2 and 6.4, we can clearly see that the average-half-face outperforms the full face in every method for the Yale Face database and the 3D face database. However, there are mixed results shown in Figure 6.3 when using the AR Face database. The best performing method for the Yale Face database was ICA with the average-half-face at 100% accuracy. For the AR Face database, the MPCA-LDA method was the best with the full faces at 91.9%. The 3D database saw the best result of 93.8% accuracy with the MPCA-LDA method and the average-half-face.

6.5.2 Bilateral Symmetry Axis Error Analysis

Finally, we performed experiments to analyze the robustness of using the average-half-face with eigenfaces to error in choosing the bilateral symme-

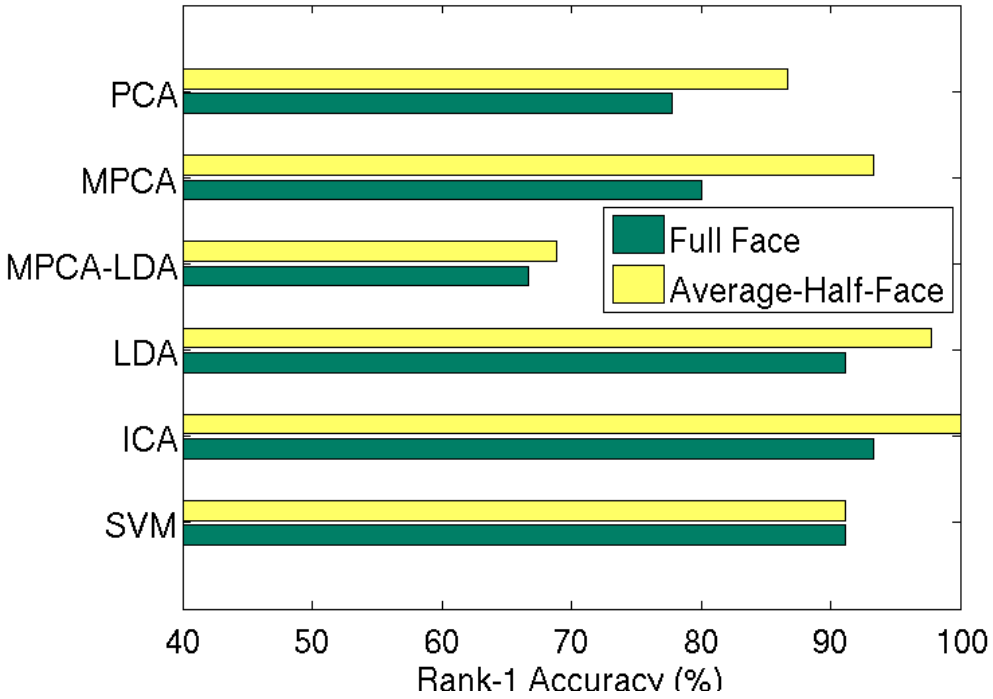


Figure 6.2: Accuracy of Full Face and Average-Half-Face on Yale Face database (A).

try axis. Let us first note that choosing the optimal axis of symmetry amounts to selecting the best vertical line in the image that divides the face image into left and right halves of the face. We perform the experiments by choosing the axis of symmetry of each image as a random (Gaussian) offset from the optimal axis of symmetry. We accomplish this by centering a Gaussian distribution at the optimum axis of symmetry with a mean of zero and variance of 5, 10, 15 and 20 pixels and sampling the desired offset from this distribution. We present the results of these experiments in Figure 6.5.

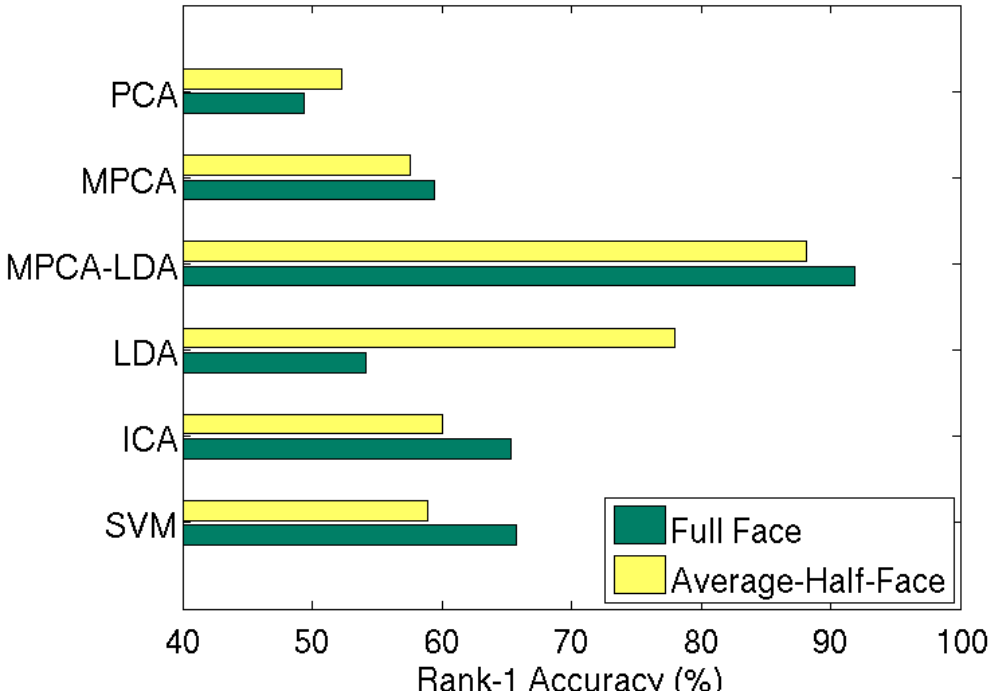


Figure 6.3: Accuracy of Full Face and Average-Half-Face on AR Face database (B).

6.6 Average-Half-Face Discussion

It is abundantly apparent from Table 6.1 that regardless of the face recognition algorithm used, utilizing the average-half-face with the Yale Face database and the 3D database produces an equal or higher accuracy rate than when using the original full face. This is not the case for every method when using the AR Face database. For instance, when using the AR Face database, the rank-1 recognition rates of the ICA and SVM methods are notably better when using the full face versus using the average-half-face. The other methods

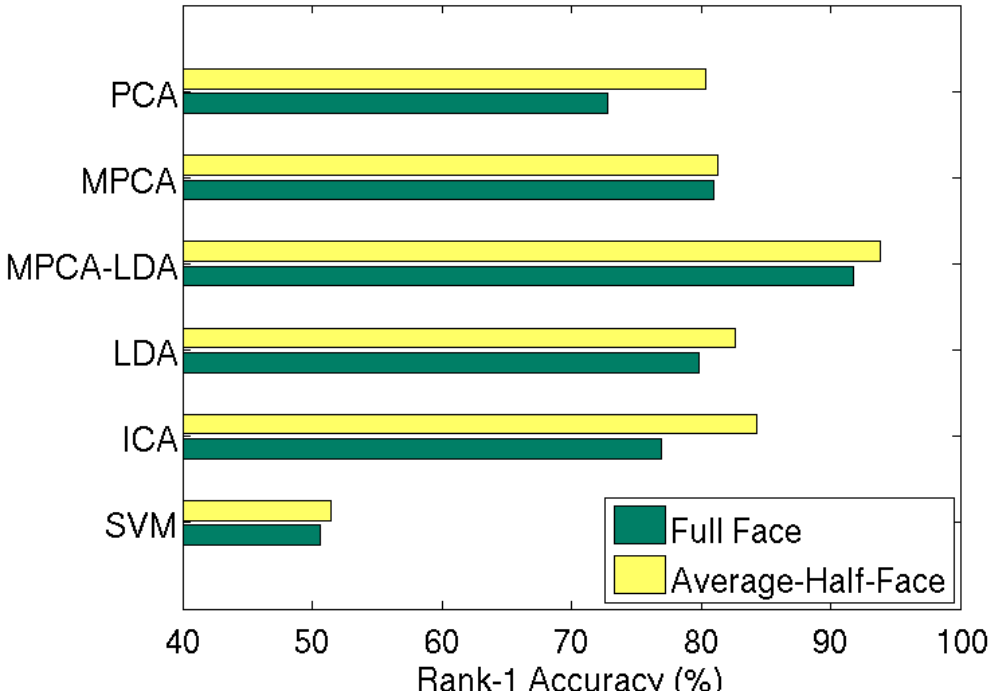


Figure 6.4: Accuracy of Full Face and Average-Half-Face on 3D Face database (C).

are very close in accuracy for the AR Face database, except for the Fisherfaces (LDA) method which shows a drastic improvement for the average-half-face. For the Yale Face database, the LDA, MPCA and eigenfaces (PCA) methods perform 6 - 13% better with the average-half-face than with the full face. All other methods with the Yale Face database are comparable, but usually have better results with the average-half-face. The 3D database gives consistently better results when using the average-half-face with a maximum accuracy increase of around 8% with the eigenfaces (PCA) method.

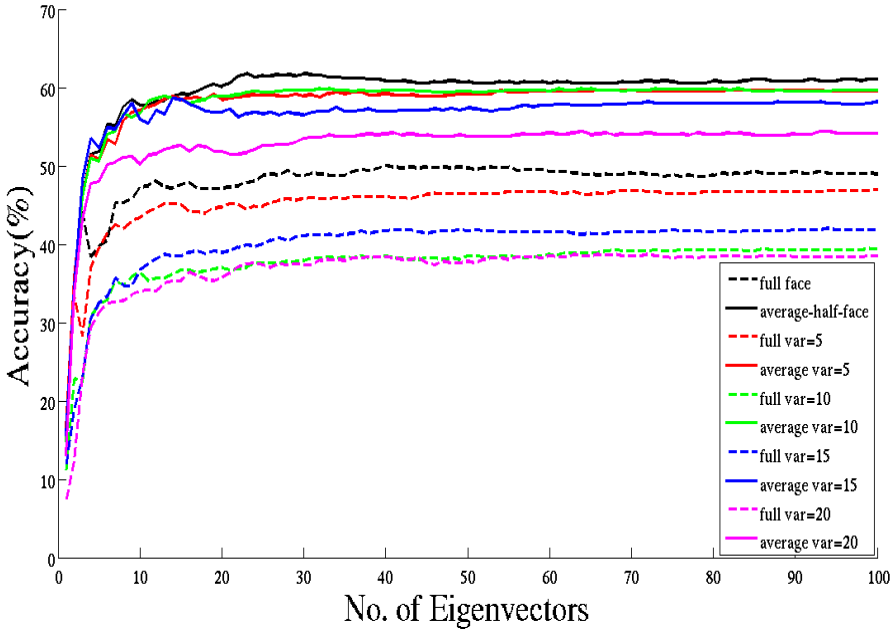


Figure 6.5: Rank 2 accuracy when choosing a suboptimal axis of symmetry.

At first glance, the results of using the AR Face database might give evidence that average-half-face is inferior to the full face. However, there are only 2 clear instances out of a total of 18 experiments that give evidence to the full face producing a higher accuracy. Therefore, the average-half-face is clearly of interest, especially since the data stored in the average-half-face is exactly half that of the full face, yet the information stored may be more discriminatory for face identification, especially in the case of the 3D database.

When considering a real world implementation of this algorithm, the noise in choosing the optimal axis of symmetry warrants consideration. From the experimental results in Figure 6.5, we can see that method of using the

average-half-face with eigenfaces is very robust to noise in choosing the optimal axis of symmetry. For most choices of the number of eigenvectors used, the performance of the system using the average-half-face with a Gaussian variance of 5 and 10 pixels is basically equivalent to using the optimal axis of symmetry. This is very promising since choosing the optimal axis of symmetry for newly acquired face images is not a trivial task. Even in the case of a variance of 15 pixels, the performance is acceptable compared to the optimal axis of symmetry. This result can be directly compared to the performance of the same experiments applied to the full face. In other words, the error in centering the full face is the same as the error in choosing the optimal axis of symmetry. Figure 6.5 displays the results of this performance. It is obvious that using the full face with eigenfaces does not result in robustness of centering the full face incorrectly. Each increase in the variance of centering the full faces results in a significant decrease in the accuracy of the method. This displays yet another advantage of using the average-half-face in 3D face recognition with eigenfaces.

The computation of the average-half-face, given the full face and the position of the middle of the face, is simple. Therefore, with a simple computation step, the accuracy of the majority of the algorithms tested was improved. We believe that this gain in accuracy has it's origin in the averaging operation, which produces a new face that contains a set of features that are more discriminatory that those of the full face. More work must be done to verify this claim and to complete the picture of the origin of this accuracy gain.

It is important to note that the results in Table 6.1 may not be the

best accuracy possible for each algorithm because some of the algorithms' parameters can be fine tuned depending on the training data set. We utilized each algorithm to compare the accuracy of using the average-half-face and the full face, since we were interested in their relative accuracies, not their absolute accuracies.

6.7 Symmetry Analysis

The above research on the average-half-face clearly shows that there may be an advantage in utilizing face symmetry to improve recognition accuracy. While promising, this work has led to several open questions. What is a good feature description or score of the symmetry of the face? Is there a statistical significance between face symmetry and face recognition? We present new symmetry scores of the face and use the scores to compare the symmetry in several subgroups of a face database. A 3D face database is used to remove the effects of illumination which should improve the reliability of the symmetry score. We find a significant difference in face symmetry between the men and women subjects in the database. The database is then partitioned into most symmetric and least symmetric subjects based on the symmetry scores. The average-half-face is utilized in our face recognition experiments to take into account the symmetry of the face. Face recognition with eigenfaces using the average-half-face is significantly higher than using the full face in all subgroups regardless of symmetry score. However, face recognition using the full face does depend on the symmetry score and generally favors the least

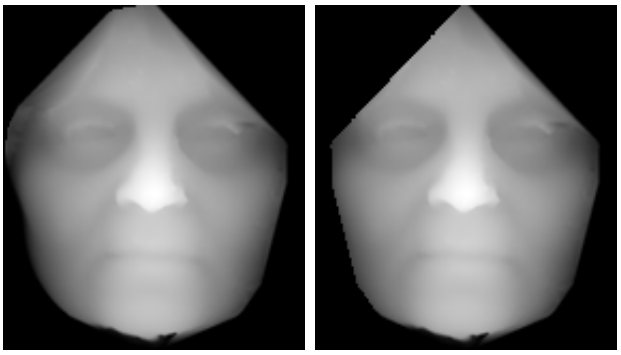
symmetric subjects.

6.7.1 3D Database Preprocessing

In our symmetry analysis experiments we have utilized a 3D face range image database known as the "Texas 3D Face Recognition Database" [29, 30, 32]. As previously explained, a 3D range image is an image in which each pixel represents the depth from the camera. The database consists of a total of 1126 images of 104 subjects. There are anywhere from 1 to 55 images per subject. For the face recognition task, we trained the algorithms using a combination of 360 images from 12 randomly chosen subjects. The gallery is formed from a single neutral expression from each of the subjects and the probes consist of the remaining images.

The images from the 3D database were preprocessed to be centered in the image using the tip of the nose location and an oval mask was applied to remove background noise. Additionally, preprocessing was performed on the images to remove any asymmetric noise that was generated during the scanning process, such as that shown in Figure 6.6 (a). An example image resulting in the above preprocessing is shown in Figure 6.6 (b).

In Figure 6.4, we can clearly see that the average-half-face outperforms the full face in every method used with the 3D face database. More information concerning these results and how they were obtained can be found in [34]. Hopefully learning the statistical differences between subjects within this database will help to uncover the difference in face recognition accuracy when



(a) (b)

Figure 6.6: (a) Example asymmetric face; (b) Preprocessed face

using the average-half-face for recognition.

6.7.2 Measuring Symmetry

Defining a measure for the symmetry of an object has been previously investigated. However, it is difficult to find a measure that encapsulates the symmetry of the face in a single number that is used to easily compare the symmetry between different faces. Authors have defined symmetry measures [64] that are useful for finding the symmetry of a single object in an image, but are not useful in comparing symmetry across images and objects, or they are not easily adapted for use with faces [22]. Some symmetry measures are based on feature points on the face and the relationships between these points, such as in [69]. However, these feature points are not readily available on every face database and require manual supervision for reliable accuracy. We have adapted one previous method for measuring symmetry, known as the density

difference (D), from the authors in [56, 62]. The measure is formulated by

$$D(i,j) = I(i,j) - I'(i,j), \tag{6.1}$$

where $I(i,j)$ is a pixel from one half of the image and $I'(i,j)$ is a pixel from the mirror of the other half of the image. The resulting density difference D is itself an image which displays the asymmetry present in the face. However, we desire a single value, or score, for the symmetry of each individual face for comparing the symmetry of many faces. Therefore, we define the following scores:

- Sum of absolute differences (s-score)
- Symmetry proportion (p-score)
- s-score applied to Gaussian smoothed image (sg-score)
- p-score applied to Gaussian smoothed image (pg-score)

The s-score is a simple extension of the density difference and is defined as:

$$s = \sum_{i,j} |D(i,j)|. \tag{6.2}$$

In addition to the s-score, we introduce a symmetry proportion score (p-score) that is bounded between 0 and 1 and may give a better intuition for the symmetry of the face. The p-score is defined as

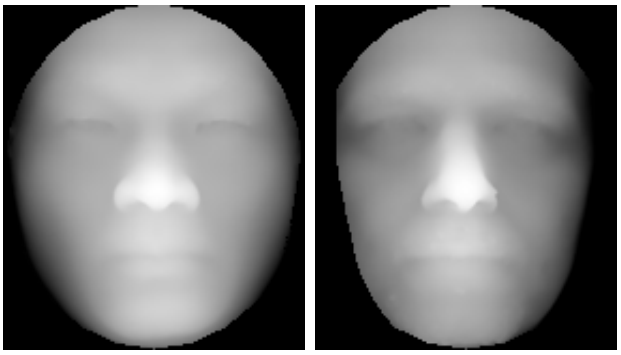
$$p = 1 - \frac{\sum_{i,j} T(i,j)}{N}, \tag{6.3}$$

where $T(i,j)$ is 0 if the absolute difference of the pixels is less than a certain threshold and 1 otherwise and N is the total number of pixels used in the symmetry score. In the experiments used in this work, the threshold chosen was 10. From this definition, it is apparent that faces that are highly symmetric will give a p-score that is close to 1.

Because these two measures are pixel based and therefore can be sensitive to noise in the image, we also apply them to a Gaussian smoothed image with a window size of 7 pixels and a sigma of 7 pixels. In the remaining sections of this chapter, the results of the scores on these smoothed images are reported as the sg-score and pg-score, respectively. These two scores may be less sensitive to errors from image alignment and the scanning process and can be thought of as comparing 7x7 patches of each side of the face. Note that the original s-score and p-score are essentially the same as the sg-score and pg-score, respectively, if the Gaussian smoothing filter has a window size of 1 pixel.

6.7.3 Statistical Analysis

As previously discussed, we wish to perform statistical analysis on the symmetry scores that we obtain from face images. For this work, we test several subgroups as follows. First we would like to test if there is a significant difference in the symmetry between men and women in the database. Sec-



(a) (b)

Figure 6.7: (a) Most symmetric and (b) least symmetric subject from the database according to symmetry scores

ond, we will use the symmetry scores of each of the subjects to partition the database (including all subjects) into the most symmetric subjects and the least symmetric subjects. This is done by using the average s-score, p-score, sg-score and pg-score of each subject and then sorting the subjects based on these average scores. The most and least symmetric subjects based on using all four symmetry scores are displayed in Figure 6.7. We will first perform tests for normality on each of the groups and then perform the appropriate hypothesis tests.

6.7.3.1 Tests for Normality

We suspect that the samples of the symmetry scores will not be normally distributed because of the upper and lower bounds on the scores. However, it is necessary to establish whether the samples are normally distributed so that the correct hypothesis test is chosen.

The first test for normality is to plot the histograms of the samples for visual inspection. Figures 6.8, 6.9, 6.10 and 6.11 display the histograms of the s-, p-, sg- and pg-scores, respectively, for the samples of men and women from the database. From the figures, it is clear that the distributions are not likely to be normal. However, this is difficult to tell from inspection alone, so we employ two statistical methods to test for normality.

The common Kolmogorov-Smirnov test [53] is first used to test normality. In all subgroups the null hypothesis that each subgroup (individually tested) was sampled from a normal distribution was rejected with an $\alpha = 0.05$. Several authors, however, have noted issues with the Kolmogorov-Smirnov test, so we have also used the recommended D'Agostino-Pearson normality test [21]. The results of this test were the same as that of the Kolmogorov-Smirnov test, so we conclude that our data is not sampled from a normal distribution. Therefore, we must take care in choosing the appropriate statistical test to discover if there is a significant difference between the subgroups.

6.7.3.2 Paired Two Sample Hypothesis Test

As indicated in the previous section, each of the subgroups that we wish to compare have been determined to not have been sampled from a normal distribution. Therefore, we cannot use statistical tests such as Student's t-test to discover if there is a significant difference between two subgroups. We have chosen to use the nonparametric Wilcoxon test [81] to test the null hypothesis that two populations have the same continuous distribution. We have utilized

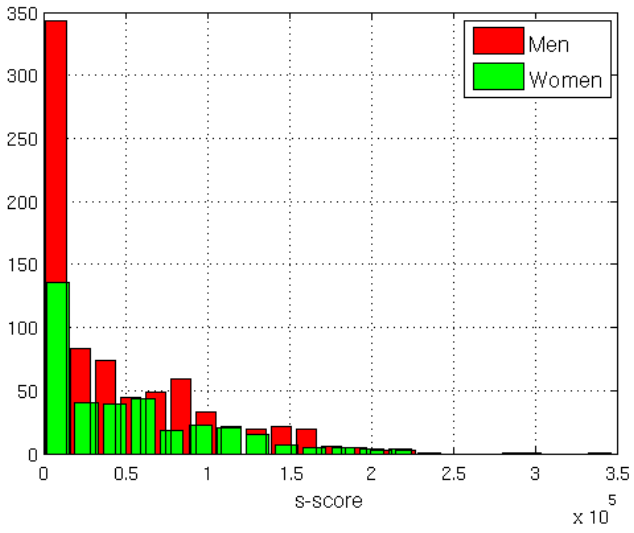


Figure 6.8: Histogram of s-score from Men and Women Images

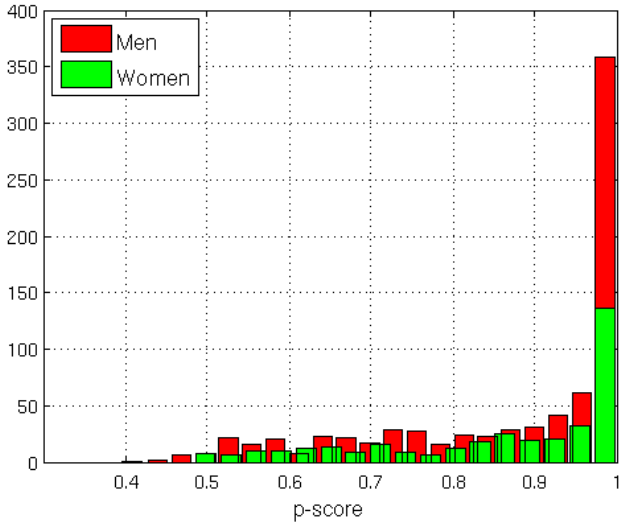


Figure 6.9: Histogram of p-score from Men and Women Images

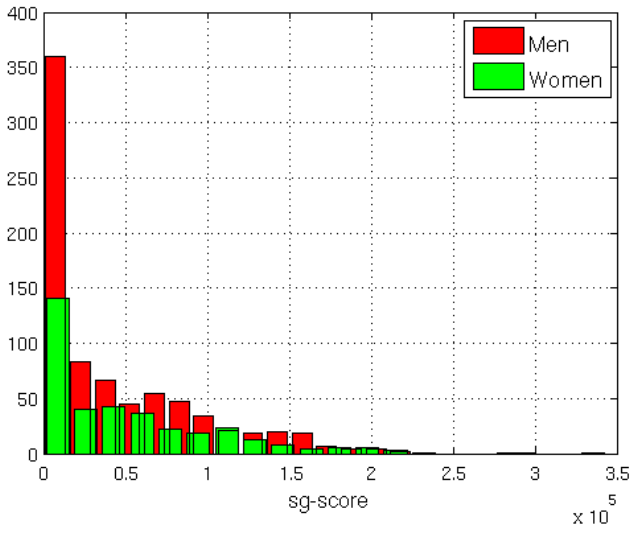


Figure 6.10: Histogram of sg-score from Men and Women Images

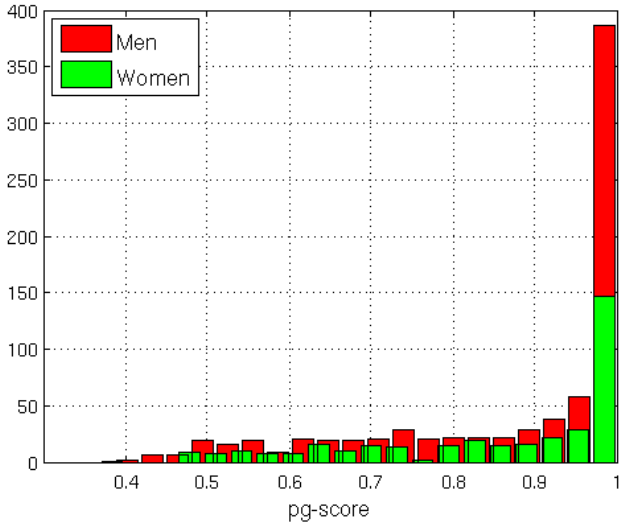


Figure 6.11: Histogram of pg-score from Men and Women Images

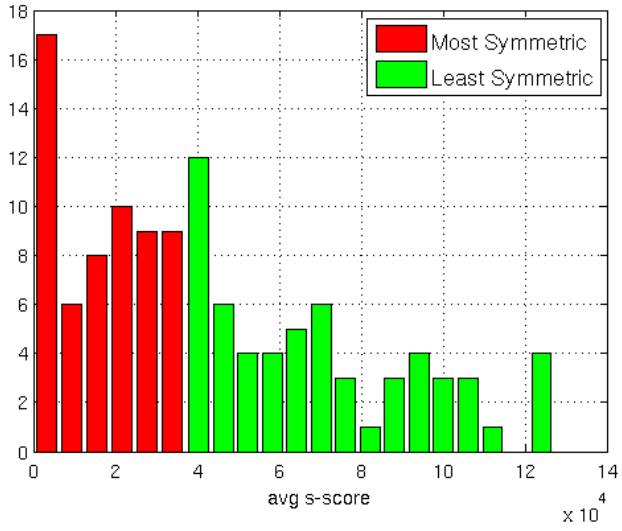


Figure 6.12: Histogram of s-score from Most and Least Symmetric Subjects

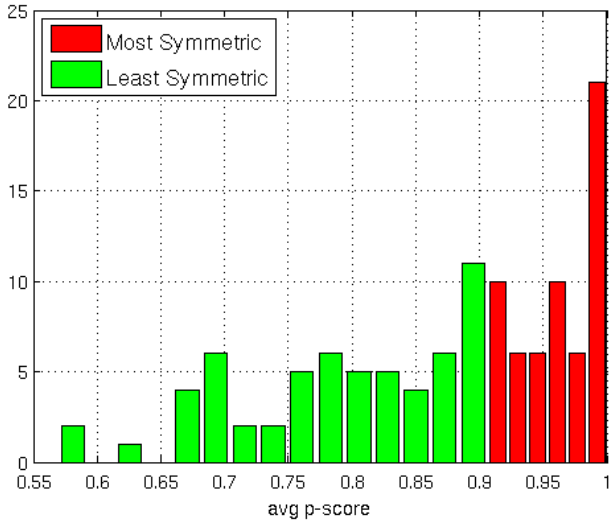


Figure 6.13: Histogram of p-score from Most and Least Symmetric Subjects

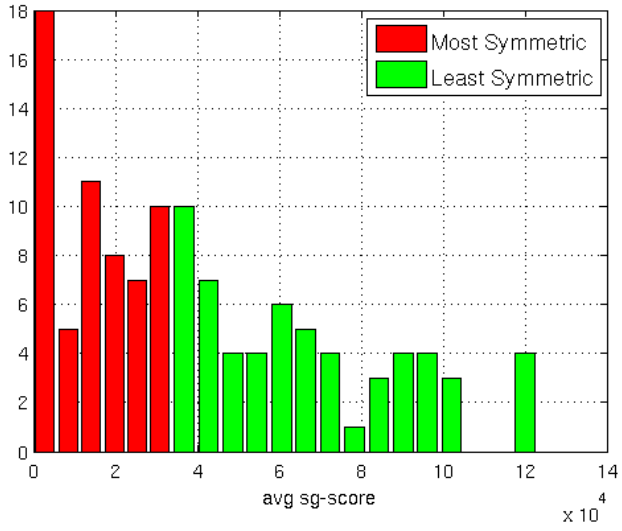


Figure 6.14: Histogram of sg-score from Most and Least Symmetric Subjects

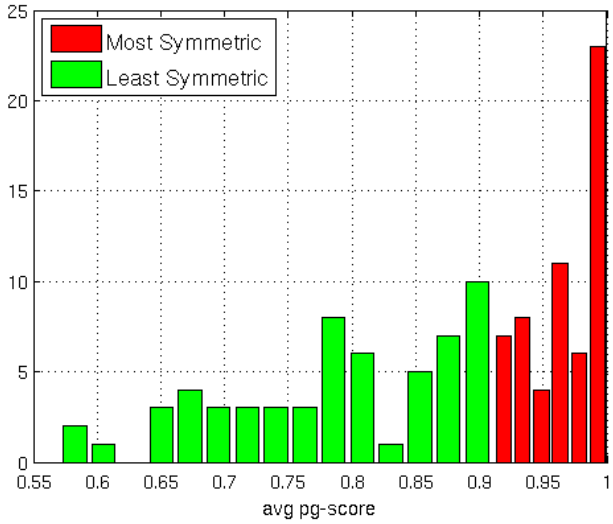


Figure 6.15: Histogram of pg-score from Most and Least Symmetric Subjects

Table 6.2: Wilcoxon Test Results

	Men v. Women		MostSym v. LeastSym	
score	h	p-val	h	p-val
s	1	0.0097	1	<0.0001
p	1	0.0257	1	<0.0001
sg	1	0.0116	1	<0.0001
pg	1	0.0206	1	<0.0001

a Matlab ® implementation of this test to produce our results.

The results have been summarized in Table 6.2. The null hypothesis in each of the tests is that the samples from each of the subgroups is drawn from the same distribution with $\alpha = 0.05$. A value of $h = 0$ means that we cannot reject the null hypothesis. A value of $h = 1$ tells us to reject the null hypothesis and conclude that the two subgroups are drawn from two significantly different distributions. The p-value all of the hypothesis tests are also included for the reader's reference.

In order to better understand this result and why a test of this type is needed, we have included the means and medians of each of the distributions in Tables 6.3 and 6.4. In a comparison such as that of men and women in the 3D database using the p-score, the means and medians seem quite close. However, as observed from Table 6.2, the subgroups are significantly different.

6.7.3.3 Face Recognition Results

We have shown that the subgroups "men" and "women" within the 3D database are sampled from two statistically different distributions. Ad-

Table 6.3: Distribution Means

	Men v. Women		MostSym v. LeastSym	
score	Men	Women	MS	LS
s	4.4E4	5.3E4	1.7E4	7.0E4
p	0.873	0.859	0.963	0.792
sg	4.2E4	5.1E4	1.5E4	6.7E4
pg	0.874	0.856	0.968	0.788

Table 6.4: Distribution Medians

	Men v. Women		MostSym v. LeastSym	
score	Men	Women	MS	LS
s	2.3E4	3.8E4	1.7E4	6.4E4
p	0.959	0.920	0.959	0.797
sg	2.0E4	3.4E4	1.6E4	6.1E4
pg	0.966	0.929	0.971	0.794

ditionally we have partitioned the database into most symmetric and least symmetric subjects according to the average s-, p-, sg- and pg-score values of each of the subjects. Now we will investigate the face recognition results of each of these subgroups independently and compare their results using both the original full face and the average-half-face.

The face recognition results are obtained in the following way using eigenfaces and nearest neighbors as the classifier. Each of the steps below is repeated separately for the full face and the average-half-face images. First, a common "face space" is formed from a random selection of 12 subjects with 30 images each and all images, including gallery and probe images, are projected into this space for recognition. The gallery is composed of 1 neutral image per subject in the subgroup and the remaining images are used as probes.

Therefore, low recognition rates are expected since each subject only has a single gallery image. Nearest neighbors is used as the classification method.

The results from the face recognition on the subgroups using both the original full face (FF) and average-half-face (AHF) are shown in Table 6.5, where the most symmetric and least symmetric subgroups are labeled as "MostSym- " and "LeastSym- " followed by the type of score used for the partitioning of the subjects, i.e., "s", "p", "sg" and "pg". The fourth column of Table 6.5 displays the p-values from the two proportion hypothesis tests which tests whether the differences in accuracy between using the full face and the average-half-face are statistically significant. The second column of Table 6.6 displays the p-values from the hypothesis test between the most and least symmetric face subgroups using the full face, while the third column displays the results when the average-half-face is used. From the results, it is clear that the average-half-face outperforms the full face for every subgroup involved.

6.7.4 Discussion

From the paired two sample hypothesis tests performed on the men and women subgroups, as well as the most symmetric and least symmetric subgroups, using the s-score, p-score, sg-score and pg-score, it is clear that each of the subgroups are sampled from statistically different distributions, as shown in Table 6.2. Therefore, one conclusion that can be drawn is that the measures of symmetry are consistent. Another is that, for this particular

Table 6.5: Face Recognition Accuracy on Subgroups

Subgroup	FF	AHF	p-value
Men	41.7	60.5	<0.0001
Women	38.2	42.5	0.4336
MostSym-s	35.0	56.3	<0.0001
LeastSym-s	46.0	54.2	0.0264
MostSym-p	42.3	60.0	<0.0001
LeastSym-p	46.4	60.9	0.0023
MostSym-sg	34.3	56.3	<0.0001
LeastSym-sg	45.5	54.0	0.0218
MostSym-pg	33.0	58.2	<0.0001
LeastSym-pg	49.9	58.0	0.0393

Table 6.6: P-values for Face Recognition Significance Between Most & Least Symmetric Subgroups

score	FF	AHF
s	0.0053	0.6614
p	0.3591	0.8650
sg	<0.0001	0.9976
pg	<0.0001	0.9817

database, the images of men are more symmetric than that of the women. When comparing the face recognition results in Table 6.5 between men and women, a similar difference is noted. When using the full face for recognition, the men have a slightly larger recognition rate than the women, though not statistically significant (p-value of 0.4419). The surprising result is that when using the symmetry of the face with the average-half-face for recognition, the men have a significantly higher recognition rate of 60.5% compared with the women's recognition rate of 42.5% (with a p-value < 0.0001). When comparing the accuracy of using the full face with the average half face for men and

women subgroups, only the result for men is statistically significant (p-value < 0.0001).

Partitioning the database into most symmetric and least symmetric subgroups with the symmetry scores produces clearly different distributions as shown in Table 6.2 and in Figures 6.12, 6.13, 6.14 and 6.15. When using these subgroups for face recognition, another surprising result is discovered. The face recognition results are higher for the least symmetric faces than that of the most symmetric faces when using the full face. This is potentially explained by thinking of a simple example. In the full face images, features that are present on only one half of the face, such as a mole or scar, are more discriminant than features that are shared on both halves of the face, so we might expect face recognition algorithms to perform better on faces which are more asymmetrical. However, the results when using the average-half-face are basically the same between the most symmetric and least symmetric halves. Of course, in the case of both symmetry scores, using the average-half-face is far more beneficial than using the full face. So, it appears that the average-half-face is not biased towards the symmetry score of the subject when performing face recognition and additionally provides a boost in accuracy to both the most symmetric and least symmetric subgroups. As shown in Table 6.5, the difference between the full face and average-half-face accuracies are statistically significant with $\alpha = 0.05$, except for the women subgroup. From Table 6.6, the difference between face recognition accuracies of the most and least symmetric subgroups are significant only for the full face results.

Therefore, when performing a face recognition task such as that described in this work, performing a face symmetry analysis on the faces in the database may help predict the face recognition performance when using a frontal full face image. However, the average-half-face appears to result in higher face recognition accuracy regardless of the symmetry inherent in the database.

6.8 Conclusion

The average-half-face has improved the accuracy, over the use of the full face, in 2D and 3D face recognition in the majority of our experiments. We have also shown that the average-half-face is robust to noise when calculating the bilateral symmetry axis of the face. These results are intriguing, but more results are needed to fully justify it's use in the recognition task.

Also, we have presented a statistical analysis of the relationship between the symmetry of the face and face recognition. We have introduced new symmetry scores and used them to compare men and women subgroups as well as most symmetric and least symmetric subgroups. We have found a statistical significance between the face symmetry of men and women subjects in the 3D database as well as differences in face recognition accuracy. The least symmetric subjects produce higher face recognition accuracy than the most symmetric subjects when using the full face. However, face recognition accuracy is universally improved when utilizing the average-half-face in our experiments over the full face.

One direction of future work is to further analyze the source of the accuracy gain of the average-half-face. We would also like to apply the average-half-face to feature extraction methods, such as those using wavelets. As seen from the results on the AR Face Database, further research into the affects of illumination, facial expressions, and occlusions on the average-half-face is needed. Further, identifying the most useful applications of the method in the face recognition field would be helpful.

The ultimate goal of this work would be to create a correlation between the symmetry of the face and face recognition that could be used to improve the overall face recognition accuracy, especially in the application of face recognition from video.

Chapter 7

Conclusion

Face recognition from video is a challenging problem. Many researchers have contributed to solving the problem from various points of view. However, an accurate and dependable solution has not yet been found.

We have presented four novel approaches to assist in face recognition from video under the following assumptions. First, we assume that the training data is a small sample size of still, mostly frontal face images. Second, the face can be successfully tracked in a single or multiple video cameras using a model-based method. Given these assumptions, we have successfully implemented our solutions to the face recognition from video problem.

First, we provide a solution to the problem in which the full face of an individual is not present in any of the frames of a video sequence. We coarsely align the face patches to a face template and then stitch the patches to reconstruct the face from a video sequence. We then use the reconstructed face in the still face recognition framework.

Next, we provide a solution to the problem of face recognition from multiple overlapping video cameras. The face is independently tracked from each of the cameras and the face texture is extracted from each of the cameras

simultaneously. Still face recognition is performed on the extracted face textures and we fuse the recognition results to obtain a significant improvement to overall face recognition accuracy. We note that the cylinder texture combined with the fusion of the recognition results attains the most accurate face recognition result. We also introduce a confidence measure of the accuracy by aggregating the results of the recognition within a particular video sequence.

We then present a novel multi-camera face tracking algorithm that utilizes the multiple views of the face to calculate a joint estimation of the face motion between frames. Camera occlusion and self-occlusion are addressed to improve the tracking result and remove errors from the tracking process. We demonstrate a significant improvement on the problem of pose estimation of the face. Additionally, the face texture generated from our tracking method is used to significantly improve the face recognition from video accuracy.

Finally, we introduce a still face recognition strategy to utilize the symmetry of the face in the face recognition process, which we call the average-half-face. We present the results of using the average-half-face on several face recognition databases and show a significant improvement in face recognition accuracy over using the original full face. We also present an analysis of the effect of face symmetry on face recognition in an attempt to discover why the average-half-face is able to perform better than the original full face.

7.1 Future Work

There are many opportunities for future work in the area of face recognition from video. By referring back to Figure 1.4, we can see that an improvement made to any one component in the face recognition from video problem has the potential to improve the final recognition result. To this end, here are the main areas we believe would benefit the research area most significantly.

First of all, the area of face tracking is extremely difficult and important for the problem of face recognition from video. In our current rigid-model based tracking approach, one deficiency in the method is the requirement of manual initialization of the model on a single frontal face image. This is a common requirement for this type of tracking approach, but automatic initialization is desired for practical implementations. Also, improvements in face tracking methods that are based on generic face models would have a large impact. This is especially true if large pose changes could be handled by a generic face model (such as AAM). Also, hybrid approaches could be used to improve the tracking accuracy as well as provide additional face texture for recognition.

Second, improvements in the area of face detection, especially for non-frontal poses, would have a significant impact on face recognition from video. The better we can detect faces and initialize the tracking of the face, the better we can track the face and produce face texture that is suitable for recognition.

Third, more analysis is needed on the average-half-face and its applica-

tion to face recognition. A clearer understanding of why the average-half-face is able to outperform the full face in many face recognition scenarios is needed. Also, further analysis of when to use the average-half-face is warranted. The ultimate goal of this research would be to predict recognition result based on the symmetry of the face and utilize the average-half-face to improve the overall accuracy of the system.

Finally, we envision a system that is built from the components introduced in this dissertation to perform automatic face recognition from multiple video cameras and is capable of handling large variations in face pose, camera occlusion and self-occlusion, and utilizes the symmetry of the face for improving the face recognition results. Such a system is the ultimate goal of this work.

On a final note, the largest area of improvement in face recognition from video is the availability of data, particularly multi-camera surveillance data. There is no doubt that the explosion of 2D still face image data available to the research community has had a profound impact on 2D still face recognition research. Hopefully the same will be true in single and multiple camera face recognition.

Bibliography

[1] Yale Univ. Face DB, 2002. `http://cvc.yale.edu/projects/yalefaces/yalefaces.html`.

[2] G. Aggarwal, A. Veeraraghavan, and R. Chellappa. 3d facial pose tracking in uncalibrated videos. *Pattern Recognition and Machine Intelligence*, pages 515–520, 2005.

[3] T. Ahonen, A. Hadid, et al. Face description with local binary patterns: Application to face recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pages 2037–2041, 2006.

[4] A.B. Ashraf, S. Lucey, and T. Chen. Learning patch correspondences for improved viewpoint invariant face recognition. *IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*, 2008.

[5] Marian Stewart Bartlett, Javier R. Movellan, and Terrence J. Sejnowski. Face recognition by independent component analysis. *IEEE Transactions on Neural Networks*, 13:1450–1464, 2002.

[6] S. Basu, I. Essa, and A. Pentland. Motion regularization for model-based head tracking. In *Pattern Recognition, 1996., Proceedings of the 13th International Conference on*, volume 3, pages 611–616. IEEE, 2002.

[7] M. Bäuml, K. Bernardin, M. Fischer, H.K. Ekenel, and R. Stiefelhagen. Multi-pose face recognition for person retrieval in camera networks. In *2010 Seventh IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS)*, pages 441–447. IEEE, 2010.

[8] Peter N. Belhumeur, João P. Hespanha, and David J. Kriegman. Eigenfaces vs. fisherfaces: Recognition using class specific linear projection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(7):711–720, 1997.

[9] A. Bhattacharyya. On a measure of divergence between two statistical populations defined by their probability distributions. *Bull. Calcutta Math. Soc*, 35(99-109):4, 1943.

[10] Jean-Yves Bouguet. Camera calibration toolbox for matlab. `http://www.vision.caltech.edu/bouguetj/calib_doc/`.

[11] G. Bradski. The OpenCV Library. *Dr. Dobb's Journal of Software Tools*, 2000.

[12] C. Bregler and J. Malik. Tracking people with twists and exponential maps. In *Computer Vision and Pattern Recognition, IEEE Computer Society Conference on*, pages 8–15. IEEE, 1998.

[13] Q. Cai, A. Sankaranarayanan, Q. Zhang, Z. Zhang, and Z. Liu. Real time head pose tracking from multiple cameras with a generic model. In *Analysis and Modeling of Faces and Gestures, (in conjunction with*

CVPR), 2010 IEEE Computer Society Workshop on, pages 25–32. IEEE, 2010.

[14] Chih-Chung Chang and Chih-Jen Lin. *LIBSVM: a library for support vector machines*, 2001. Software available at `http://www.csie.ntu.edu.tw/~cjlin/libsvm`.

[15] R. Chellappa, P. Sinha, and P.J. Phillips. Face recognition by computers and humans. *Computer*, 43(2):46–55, 2010.

[16] R. Chellappa, S. Zhou, and B. Li. Bayesian methods for face recognition from video. In *Acoustics, Speech, and Signal Processing (ICASSP), 2002 IEEE International Conference on*, volume 4, pages IV–4068. IEEE, 2002.

[17] Xin Chen, Patrick J. Flynn, and Kevin W. Bowyer. Fully Automated Facial Symmetry Axis Detection in Frontal Color Images. In *Automatic Identification Advanced Technologies, 2005. Fourth IEEE Workshop on*, pages 106–111. IEEE, 2005.

[18] Z.W. Chen, C.C. Chiang, and Z.T. Hsieh. Extending 3d lucas–kanade tracking with adaptive templates for head pose estimation. *Machine Vision and Applications*, 21(6):889–903, 2010.

[19] S. Choi and D. Kim. Robust head tracking using 3D ellipsoidal head model in particle filter. *Pattern Recognition*, 41(9):2901–2915, 2008.

[20] Ciarán O. Conaire, Noel E. O'Connor, and Alan F. Smeaton. Detector adaptation by maximising agreement between independent data sources. In *CVPR*. IEEE Computer Society, 2007.

[21] RB D'Agostino. Tests for normal distribution. *Goodness-Of-Fit Techniques*, pages 367–419, 1986.

[22] S.C. Dakin and A.M. Herbert. The spatial region of integration for visual symmetry detection. *Proceedings of the Royal Society B: Biological Sciences*, 265(1397):659, 1998.

[23] T.A. Davis and C. Ramanujacharyulu. Statistical analysis of bilateral symmetry in plant organs. *Sankhyā: The Indian Journal of Statistics, Series B*, 33(3):259–290, 1971.

[24] R.O. Duda, P.E. Hart, and D.G. Stork. *Pattern classification*, volume 2. John Wiley & Sons, New York, 2001.

[25] P.K. Endress. Symmetry in flowers: diversity and evolution. *International Journal of Plant Sciences*, 160(6):3–23, 1999.

[26] I.A. Essa and A.P. Pentland. Coding, analysis, interpretation, and recognition of facial expressions. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 19(7):757–763, 2002.

[27] N. Faggian, A. Paplinski, and T.J. Chin. Face recognition from video using active appearance model segmentation. In *Pattern Recognition,*

2006. ICPR 2006. 18th International Conference on, volume 1, pages 287–290. IEEE, 2006.

[28] Z. Guo, L. Zhang, D. Zhang, and X. Mou. Hierarchical multiscale lbp for face and palmprint recognition. In *Image Processing, 17th IEEE International Conference on on*, pages 4521–4524. IEEE, 2010.

[29] S. Gupta, K. R. Castleman, M. K. Markey, and A. C. Bovik. Texas 3D Face Recognition Database. *IEEE Southwest Symposium on Image Analysis and Interpretation*, pages 97–100, May 2010.

[30] S. Gupta, K. R. Castleman, M. K. Markey, and A. C. Bovik. *Texas 3D Face Recognition Database*, 2010. `http://live.ece.utexas.edu/research/texas3dfr/index.htm`.

[31] S. Gupta, M.K. Markey, and A.C. Bovik. Anthropometric 3d face recognition. *International journal of computer vision*, 90(3):331–349, 2010.

[32] Shalini Gupta, J. K. Aggarwal, Mia K. Markey, and Alan C. Bovik. 3D face recognition founded on the structural diversity of human faces. *Computer Vision and Pattern Recognition, IEEE Computer Society Conference on*, 0:1–7, 2007.

[33] Josh Harguess. Full-motion recovery from multiple video cameras: `http://cvrc.ece.utexas.edu/research/vs2011`. August 2011.

[34] Josh Harguess and J. K. Aggarwal. A case for the average-half-face in 2D and 3D for face recognition. In *IEEE Computer Society Workshop on Biometrics (in conjunction with CVPR)*, June 2009.

[35] Josh Harguess and J. K. Aggarwal. The Average-Half-Face in 2D and 3D Face Recognition. In *Pattern Recognition and Machine Vision*, pages 135–148. River Publishers, 2010.

[36] Josh Harguess, Shalini Gupta, and J. K. Aggarwal. 3D face recognition with the average-half-face. In *ICPR*, pages 1–4. IEEE, December 2008.

[37] Josh Harguess, Changbo Hu, and J. K. Aggarwal. Fusing face recognition from multiple cameras. *Workshop on Applications of Computer Vision (WACV)*, 2009.

[38] Josh Harguess, Changbo Hu, and J. K. Aggarwal. Full-motion recovery from multiple video cameras applied to face tracking and recognition. *The Eleventh IEEE International Workshop on Visual Surveillance*, November 2011.

[39] Josh Harguess, Changbo Hu, and J. K. Aggarwal. Occlusion robust multi-camera face tracking. *The 3rd International Workshop on Machine Learning for Vision-based Motion Analysis (MLvMA-2011) in conjunction with IEEE CVPR 2011*, June 2011.

[40] J. Heo and M. Savvides. Face recognition across pose using view based active appearance models (vbaams) on cmu multi-pie dataset. *Computer*

Vision Systems, pages 527–535, 2008.

[41] J. Honekopp, T. Bartholome, and G. Jansen. Facial attractiveness, symmetry, and physical fitness in young women. *Human Nature*, 15(2):147–167, 2004.

[42] Changbo Hu, Josh Harguess, and J. K. Aggarwal. Patch-based face recognition from video. In *International Conference on Image Processing (ICIP)*, pages 1–4, November 2009.

[43] G.B. Huang, M. Ramesh, T. Berg, and E. Learned-Miller. Labeled faces in the wild: A database for studying face recognition in unconstrained environments. *University of Massachusetts, Amherst, Technical Report 07*, 49:1, 2007.

[44] H. Huo and J. Feng. Face recognition via aam and multi-features fusion on riemannian manifolds. *Computer Vision–ACCV 2009*, pages 591–600, 2009.

[45] T.S. Jebara and A. Pentland. Parametrized structure from motion for 3d adaptive feedback tracking of faces. In *cvpr*, page 144. Published by the IEEE Computer Society, 1997.

[46] P. Jiménez, J. Nuevo, and L.M. Bergasa. Face pose estimation and tracking using automatic 3D model construction. *Computer Vision and Pattern Recognition Workshops (CVPRW), 2008 IEEE Computer Society Conference on*, pages 1–7, 2008.

[47] N. Kumar, A. Berg, P. Belhumeur, and S. Nayar. Describable visual attributes for face verification and image search. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, (99):1–1, 2011.

[48] M. La Cascia, S. Sclaroff, and V. Athitsos. Fast, reliable head tracking under varying illumination: An approach based on registration of texture-mapped 3d models. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 22(4):322–336, 2000.

[49] Hyung-Soo Lee and Daijin Kim. Tensor-based aam with continuous variation estimation: Application to variation-robust face recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 31:1102–1116, 2009.

[50] Kuang-Chih Lee, Jeffrey Ho, Ming-Hsuan Yang, and David Kriegman. Video-based face recognition using probabilistic appearance manifolds. *IEEE Conf. On Computer Vision and Pattern Recognition*, 1:313–320, 2003.

[51] Y. Li, S. Gong, and H. Liddell. Constructing facial identity surfaces for recognition. *International Journal of Computer Vision*, 53(1):71–92, 2003.

[52] S. Liao, X. Zhu, Z. Lei, L. Zhang, and S. Li. Learning multi-scale block local binary patterns for face recognition. *Advances in Biometrics*, pages 828–837, 2007.

[53] H.W. Lilliefors. On the Kolmogorov-Smirnov test for normality with mean and variance unknown. *Journal of the American Statistical Association*, 62(318):399–402, 1967.

[54] D.T. Lin and M.J. Liu. Face occlusion detection for automated teller machine surveillance. *Lecture notes in computer science*, 4319:641–651, 2006.

[55] X. Liu and T. Cheng. Video-based face recognition using adaptive hidden markov models. In *Computer Vision and Pattern Recognition, 2003. Proceedings. 2003 IEEE Computer Society Conference on*, volume 1, pages I–340. IEEE, 2003.

[56] Y. Liu, K.L. Schmidt, J.F. Cohn, and S. Mitra. Facial asymmetry quantification for expression invariant human identification. *Computer Vision and Image Understanding*, 91(1-2):138–159, 2003.

[57] Z. Liu and Z. Zhang. Robust head motion computation by taking advantage of physical properties. In *Human Motion, 2000. Proceedings. Workshop on*, pages 73–77. IEEE, 2002.

[58] D.G. Lowe. Robust model-based motion tracking through the integration of search and estimation. *International Journal of Computer Vision*, 8(2):113–122, 1992.

[59] Haiping Lu, Konstantinos N. Plataniotis, and Anastasios N. Venetsanopoulos. Mpca: Multilinear principal component analysis of tensor objects.

IEEE Trans. on Neural Networks, 19(1):18–39, 2008.

[60] Bruce D. Lucas and Takeo Kanade. An iterative image registration technique with an application to stereo vision. *IEEE Proceedings of the 7th International Joint Conference on Artificial Intelligence*, April, 1981, pp. 674-679.

[61] A.M. Martinez and R. Benavente. The AR face database. *CVC Technical Report*, (24), June 1998.

[62] S. Mitra, N.A. Lazar, and Y. Liu. Understanding the role of facial asymmetry in human face identification. *Statistics and Computing*, 17(1):57–70, 2007.

[63] Erik Murphy-Chutorian and Mohan Manubhai Trivedi. Head pose estimation in computer vision: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 31:607–626, 2009.

[64] D. O'Mara and R. Owens. Measuring bilateral symmetry in digital images. In *TENCON'96. Proceedings. 1996 IEEE TENCON. Digital Signal Processing Applications*, volume 1, pages 151–156. IEEE, 2002.

[65] Gang Pan and Zhaohui Wu. 3D face recognition from range data. *Int. J. Image Graphics*, 5(3):573–594, 2005.

[66] U. Park, A.K. Jain, and A. Ross. Face recognition in video: Adaptive fusion of multiple matchers. In *Computer Vision and Pattern Recognition, 2007. CVPR'07. IEEE Conference on*, pages 1–8. IEEE, 2007.

[67] Narayanan Ramanathan. Facial similarity across age, disguise, illumination and pose. In *Proceedings of International Conference on Image Processing*, 1999.

[68] G. Rhodes, F. Proffitt, J.M. Grady, and A. Sumich. Facial symmetry and the perception of beauty. *Psychonomic Bulletin and Review*, 5:659–669, 1998.

[69] K. Schmid, D. Marx, and A. Samal. Computation of a face attractiveness index based on neoclassical canons, symmetry, and golden ratios. *Pattern Recognition*, 41(8):2710–2717, 2008.

[70] Bernhard Schölkopf, Alex J. Smola, Robert C. Williamson, and Peter L. Bartlett. New support vector algorithms. *Neural Comput.*, 12(5):1207–1245, 2000.

[71] J. See and C. Eswaran. Exemplar extraction using spatio-temporal hierarchical agglomerative clustering for face recognition in video. *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2011.

[72] Mili I. Shah and Danny C. Sorensen. A symmetry preserving singular value decomposition. *SIAM J. Matrix Anal. Appl.*, 28(3):749–769, 2006.

[73] G. Shakhnarovich, L. Lee, and T. Darrell. Integrated face and gait recognition from multiple views. *Computer Vision and Pattern Recognition (CVPR). IEEE Conference on*, 2001.

[74] Gregory Shakhnarovich and Baback Moghaddam. Face recognition in subspaces. In *S.Z. Li, A.K. Jain (Eds.), Handbook of Face Recognition*, pages 141–168. Springer, 2004.

[75] Scott Stillman, Rawesak Tanawongsuwan, and Irfan Essa. A system for tracking and recognizing multiple people with multiple cameras. In *In Proceedings of Second International Conference on Audio-Visionbased Person Authentication*, pages 96–101, 1998.

[76] Jaewon Sung, Takeo Kanade, and Daijin Kim. Pose robust face tracking by combining active appearance models and cylinder head models. *International Journal of Computer Vision*, 80(2):260–274, 2008.

[77] L. Teijeiro-Mosquera, J.L. Alba-Castro, and D. Gonzalez-Jimenez. Face recognition across pose with automatic estimation of pose parameters through aam-based landmarking. In *2010 International Conference on Pattern Recognition*, pages 1339–1342. IEEE, 2010.

[78] R. Thornhill and S.W. Gangestad. Facial attractiveness. *Trends in Cognitive Sciences*, 3(12):452–460, 1999.

[79] M. A. Turk and A. P. Pentland. Eigenfaces for recognition. *Journal of Cognitive Neuroscience*, 3(1):71–86, 1991.

[80] H. Wang, Y. Wang, and Y. Cao. Video-based face recognition: A survey. *World Academy of Science, Engineering and Technology*, 60:293–302, 2009.

[81] F. Wilcoxon. Individual comparisons by ranking methods. *Biometrics Bulletin*, 1(6):80–83, 1945.

[82] L. Wiskott, J.M. Fellous, N. Kuiger, and C. von der Malsburg. Face recognition by elastic bunch graph matching. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 19(7):775–779, 1997.

[83] Y. Wong, S. Chen, S. Mau, C. Sanderson, and B.C. Lovell. ChokePoint Dataset, 2011. `http://itee.uq.edu.au/~uqywong6/chokepoint.html`.

[84] Y. Wong, S. Chen, S. Mau, C. Sanderson, and B.C. Lovell. Patch-based probabilistic image quality assessment for face selection and improved video-based face recognition. *IEEE Biometrics Workshop, Computer Vision and Pattern Recognition (CVPR) Workshops*, pages 81–88, 2011.

[85] John Wright, Allen Yang, Arvind Ganesh, Shankar Sastry, and Yi Ma. Robust face recognition via sparse representation. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 31(2):210–227, 2009.

[86] Jing Xiao, Tsuyoshi Moriyama, Takeo Kanade, and Jeffrey Cohn. Robust full-motion recovery of head by dynamic templates and re-registration techniques. *International Journal of Imaging Systems and Technology*, 13:85 – 94, September 2003.

[87] Binglong Xie, Terry Boult, Visvanathan Ramesh, and Ying Zhu. Multi-camera face recognition by reliability-based selection. In *IEEE Interna-*

tional Conference on Computational Intelligence for Homeland Security and Personal Safety, pages 18–23, October 2006.

[88] Binglong Xie, Visvanathan Ramesh, Ying Zhu, and Terrance E. Boult. On channel reliability measure training for multi-camera face recognition. In *WACV*, page 41. IEEE Computer Society, 2007.

[89] G. Zhang, X. Huang, S. Li, Y. Wang, and X. Wu. Boosting local binary pattern (lbp)-based face recognition. *Advances in biometric person authentication*, pages 179–186, 2005.

[90] Liyan Zhang, Anshuman Razdan, Gerald Farin, John Femiani, MyungSoo Bae, and Charles Lockwood. 3D face authentication and recognition based on bilateral symmetry analysis. *The Visual Computer*, 22(1):43–55, 2006.

[91] Y. Zhang and C. Kambhamettu. 3D head tracking under partial occlusion. *Pattern Recognition*, 35(7):1545–1558, 2002.

[92] Zhengyou Zhang, Zicheng Liu, Dennis Adler, Michael F. Cohen, Erik Hanson, and Ying Shan. Robust and rapid generation of animated faces from video images. *International Journal of Computer Vision*, 58(2):93–119, 2004.

[93] W. Zhao, R. Chellappa, P.J. Phillips, and A. Rosenfeld. Face recognition: A literature survey. *Acm Computing Surveys (CSUR)*, 35(4):399–458, 2003.

[94] WenYi Zhao and Rama Chellappa. Illumination-insensitive face recognition using symmetric shape-from-shading. *Computer Vision and Pattern Recognition, IEEE Computer Society Conference on*, 1:1286, 2000.

[95] S. Zhou, V. Krueger, and R. Chellappa. Probabilistic recognition of human faces from video. *Computer Vision and Image Understanding*, 91:214–245, 2003.